Medical Inventory Management

COLLEGE NAME : Sri Krishna Adithya College of Arts and Science
COLLEGE CODE : BRUAG
TEAM ID : NM2025TMID22165
TEAM SIZE : 4
TEAM LEADER : KARTHIKEYAN E
Email: 23bsit129karthikeyane@skacas.ac.in
Team Member 1: KAVITHARAN
Email : 23bsit130kavitharanm@skacas.ac.in
Team Member 2: MADHUMITHA V
Email: 23bsit136madhumithav@skacas.ac.in
Team Member 3: MARIYAM BEEVI D
Email: 23bsit137mariyambeevid@skacas.ac.in

Title: Medical Inventory Management

Project Overview

The **Medical Inventory Management System** is a Salesforce CRM-based application created to handle medical supplies, medicines, and equipment with accuracy and efficiency. It simplifies operations such as purchase order handling, supplier management, and inventory tracking. By automating these processes, the system lowers manual effort, reduces mistakes, and ensures that healthcare organizations always have essential medical resources available.

Objectives

The system has been developed with the following goals:

- Keep all medical inventory information in a single, centralized place.
- Make the purchase order process faster through automation and approvals.
- Record supplier details and track their order history for better accountability.
- Alert users whenever stock is low or products are nearing expiry.
- Provide managers with useful reports and dashboards for smarter decision-making.
- Apply Salesforce features like workflows, validation rules, and triggers.
- Ensure data security by giving different access levels to Admins, Inventory Managers, and Staff members.

Scope of the Project

What is included:

- Tracking of medicines and medical equipment.
- Supplier records and purchase order processing.
- Role-based security for users.
- Salesforce automation (triggers, workflows, approval processes).
- Reports and dashboards for monitoring and insights.

What is not included:

- Billing or payment system integration with hospitals.
- Advanced demand forecasting using AI/ML.
- Offline mobile application development.

Key Modules

1. **Inventory Management** – Add, update, and view medical items.
2. **Supplier Management** – Save supplier details and monitor past orders.
3. **Purchase Order Management** – Automate creation, approvals, and cost calculations.
4. **Stock Monitoring** – Notifications for low quantity or expired items.
5. **Reports & Dashboards** – Summaries and visual reports for managers.
6. **User Management** – Roles and profiles for secure access.

Benefits

- Reduced wastage and better stock control.
- Less manual effort in creating and approving purchase orders.
- Stronger supplier accountability.
- Real-time access to inventory and cost details.

Student Learning Outcomes

1. **Knowledge of Salesforce CRM** – Hands-on customization for practical applications.
2. **Cloud Development Skills** – Creating custom fields, objects, layouts, and relationships.
3. **Business Automation Experience** – Building triggers, workflows, and approval processes.
4. **Data Analytics Skills** – Developing dashboards and reports for inventory insights.
5. **Security Management** – Implementing user profiles with different access permissions.

6. **Problem-Solving Ability** – Addressing real challenges in healthcare inventory.
7. **Documentation & Presentation Skills** – Preparing clear technical reports.
8. **Collaboration & Project Management** – Working effectively with peers to meet deadlines (for team projects).

System Requirements

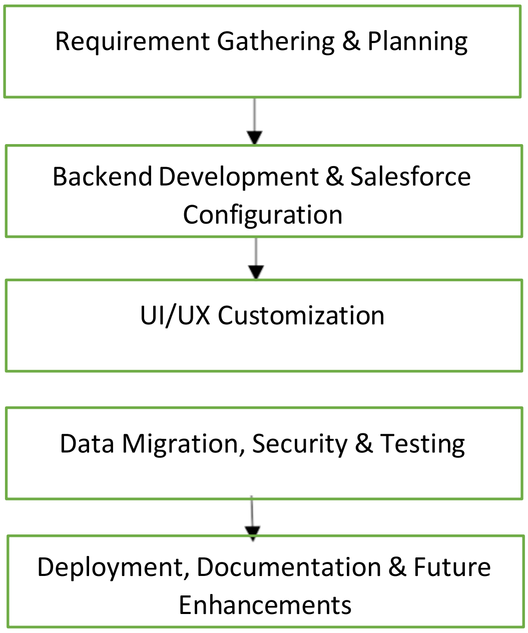
Hardware

- **Processor:** Intel Core i3 / AMD equivalent or higher
- **RAM:** 4 GB minimum (8 GB preferred)
- **Storage:** At least 20 GB free space
- **Display:** 1024×768 or higher
- **Connectivity:** Reliable internet connection

Software

- **OS:** Windows 10/11, Linux, or macOS
- **Browser:** Chrome, Firefox, or Edge (latest versions)
- **Salesforce Edition:** Developer Edition (*Naan Mudhalvan provided*)
- **Optional Tools:** Developer Console, VS Code with Salesforce extensions
- **Documentation Tools:** MS Office, Google Docs

PROJECT PHASES


Requirement Gathering & Planning
Backend Development & Salesforce Configuration
UI/UX Customization
Data Migration, Security & Testing
Deployment, Documentation & Future Enhancements

Phase 1: Requirement Analysis & Planning

-Clear understanding of the requirements collected from healthcare staff, administrators, and inventory managers.

-Well-defined project objectives and scope.

-Drafted preliminary data model and workflow diagrams for suppliers, purchase orders, and medical items.

-Structured project roadmap prepared for smooth execution in later phases.

Phase 2: Salesforce Development – Backend & Configurations

Milestone 1: Salesforce Developer Account Setup

Activity 1: Creating a Developer Account

-Registered for a Salesforce Developer Edition account through the official signup page:
https://developer.salesforce.com/signup

-Configured the developer org to serve as the working environment for backend customization and application development.

Milestone 2: Creating Custom Objects

Activity 1: Creating a Product Object

Steps to create the custom object *Product*:

-Navigate to Setup in Salesforce.

-Select Object Manager from the menu.

-Click Create → Custom Object.

Provide the following details:

-Label Name: Product

-Plural Label: Products

-Record Name: Product ID

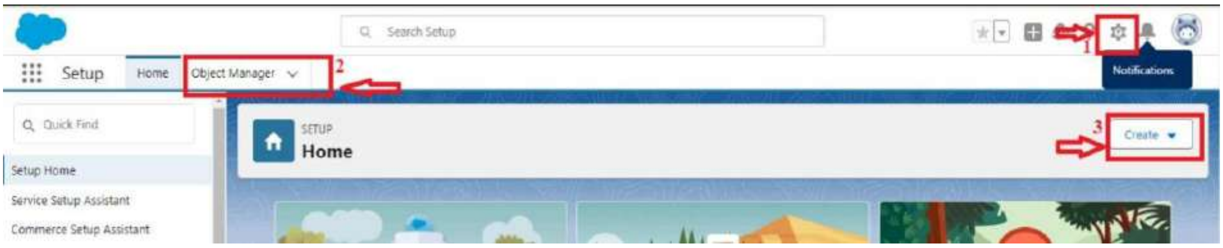
-Data Type: Text

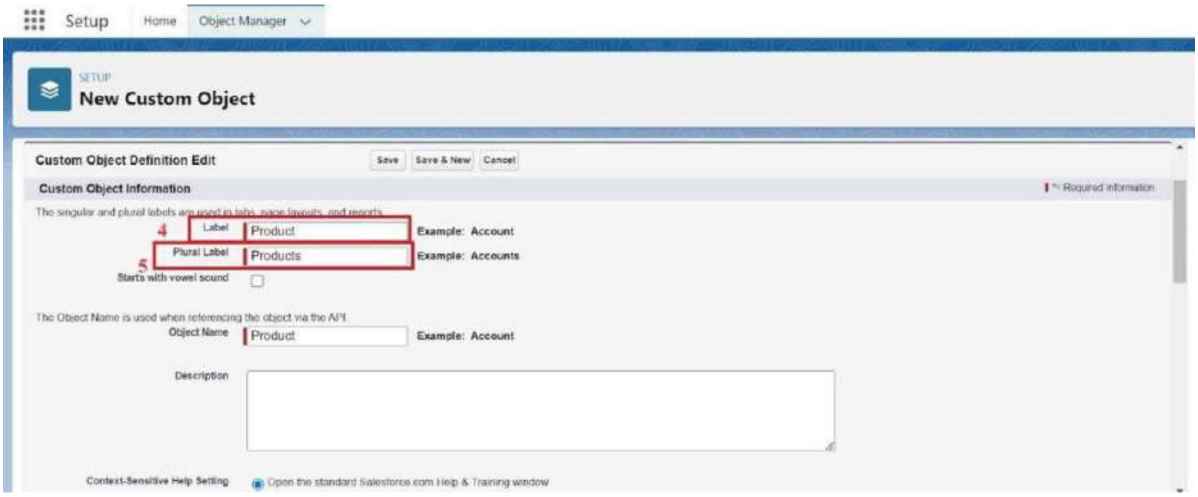
Enable the following options:

-Allow Reports

-Allow Search

-Click Save and New to complete the process.





New Custom Object

Custom Object Definition Edit

Custom Object Information

| = Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label: Product Example: Account

Plural Label: Products Example: Accounts

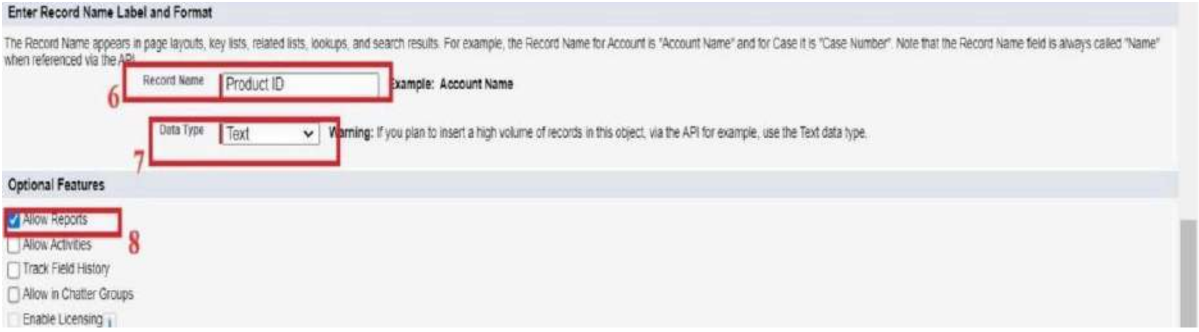
Starts with vowel sound □

The Object Name is used when referencing the object via the API.

Object Name: Product Example: Account

Description

Context-Sensitive Help Setting ● Open the standard Salesforce.com Help & Training window



Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name: Product ID Example: Account Name

Data Type: Text Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Optional Features

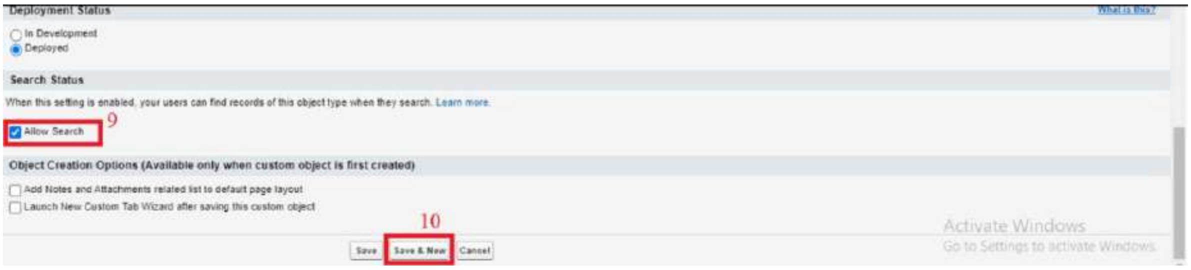
☑ Allow Reports

□ Allow Activities

□ Track Field History

□ Allow in Chatter Groups

□ Enable Licensing



Deployment Status

○ In Development

● Deployed

Search Status

When this setting is enabled, your users can find records of this object type when they search. Learn more.

☑ Allow Search

Object Creation Options (Available only when custom object is first created)

□ Add Notes and Attachments related list to default page layout

□ Launch New Custom Tab Wizard after saving this custom object

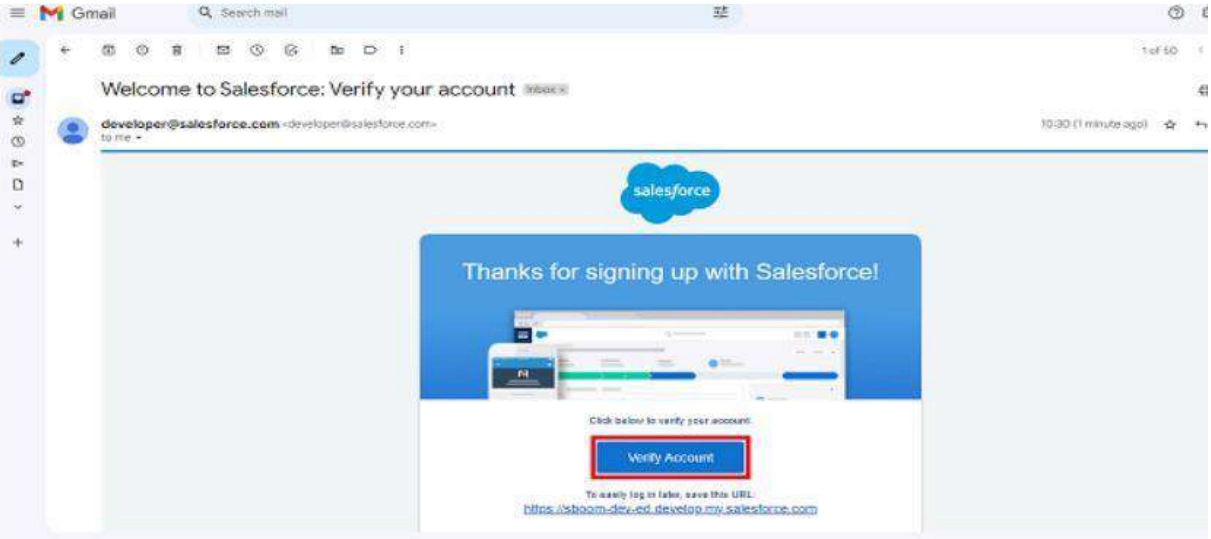
Activity 2: Account Activation

Steps to verify and activate the Salesforce Developer Account:

-Open the inbox of the email address used during signup and look for the verification email (may take 5–10 minutes to arrive).

-Click on the “Verify Account” link provided in the email.

-Your Salesforce Developer account will be successfully activated and ready for use.



Welcome to Salesforce: Verify your account

developer@salesforce.com <developer@salesforce.com>
to me
10:30 (1 minute ago)

Thanks for signing up with Salesforce!

Click below to verify your account.

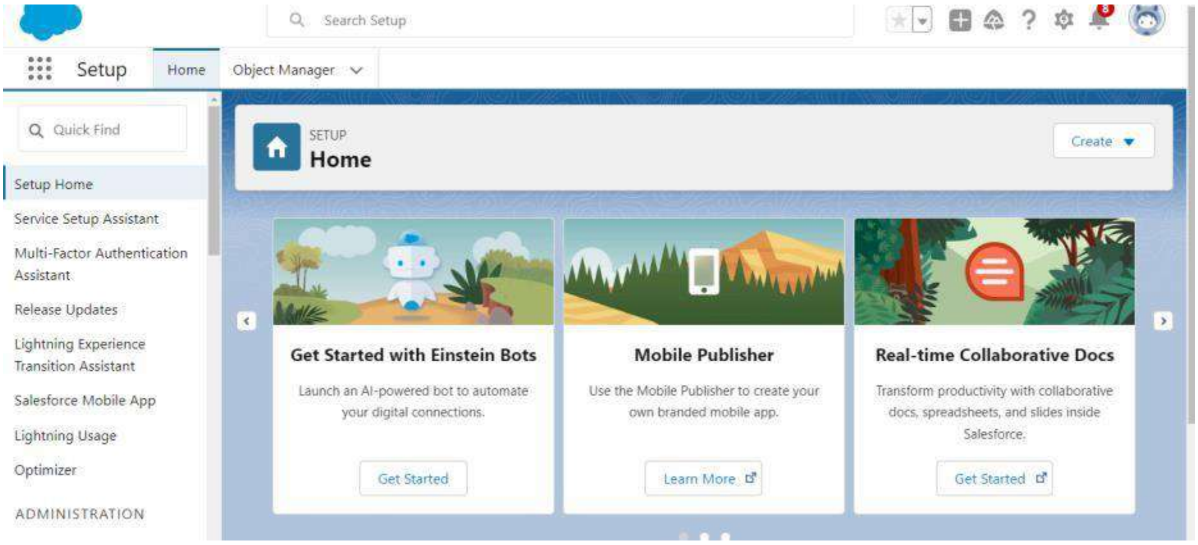
To easily log in later, save this URL:
https://sboom-dev-ed.develop.my.salesforce.com

-Open the inbox of the email address used during signup (the email may take 5–10 minutes to arrive).

-Click on the “Verify Account” link provided in the email.

-Set a new password, choose and answer a security question, then click Change Password.

-After completing the setup, you will be redirected to your Salesforce Setup Home Page.



Get Started with Einstein Bots

Launch an AI-powered bot to automate your digital connections.

Mobile Publisher

Use the Mobile Publisher to create your own branded mobile app.

Real-time Collaborative Docs

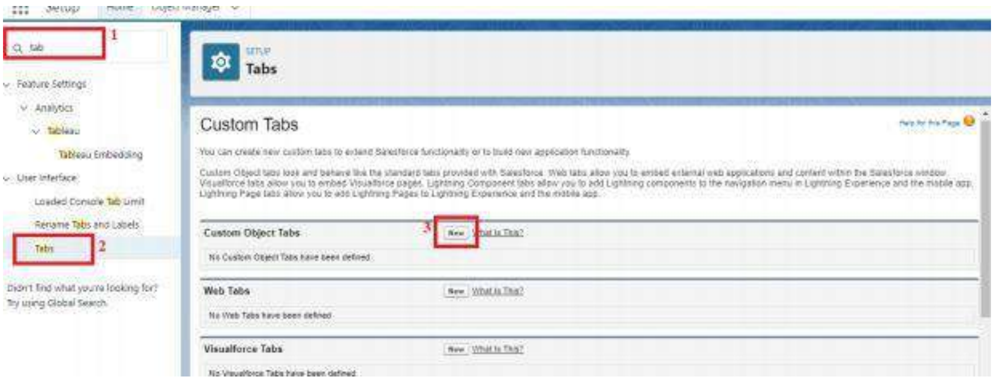
Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce.

Milestone 3: Creating Tabs

Activity 1: Creating a Tab for the Product Object

Procedure:

1. Navigate to **Setup** and type **Tabs** in the Quick Find search bar.
2. Select **Tabs** from the available options.
3. Under **Custom Object Tabs**, click **New**.
4. From the list, choose **Product** as the object and select an appropriate **Tab Style**.
5. Click **Next**. On the **Add to Profiles** page, retain the default settings and click **Next** again.
6. On the **Add to Custom Apps** page, uncheck the option **Include Tab**.
7. Make sure the option **Append tab to user's existing personal customizations** is selected.
8. Finally, click **Save** to complete the tab creation process.



Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs

No Custom Object Tabs have been defined.

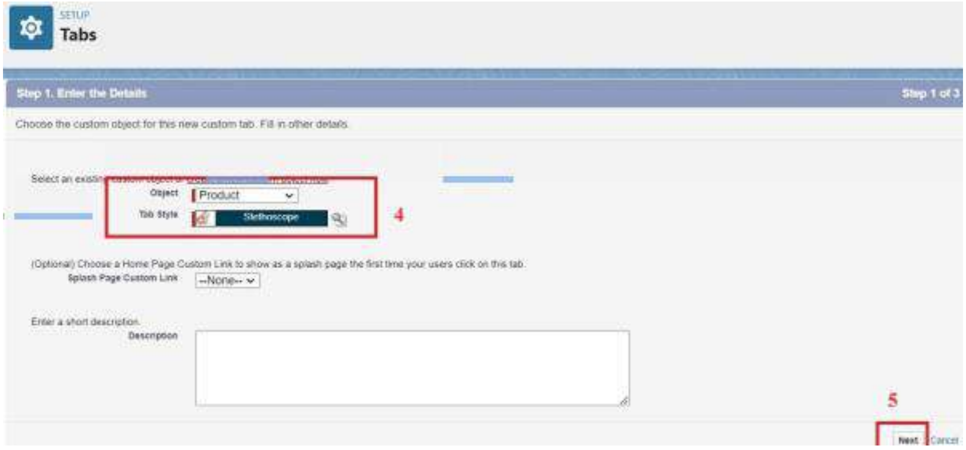
Web Tabs

No Web Tabs have been defined.

Visualforce Tabs

No Visualforce Tabs have been defined.

Didn't find what you're looking for? Try using Global Search.



Step 1. Enter the Details

Step 1 of 3

Choose the custom object for this new custom tab. Fill in other details.

Select an existing custom object or create a new one now.

Object: Product

Tab Style: Stethoscope

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

Splash Page Custom Link: --None--

Enter a short description.

Description

Milestone 3: Creating Tabs

Activity 2: Creating Remaining Tabs

Procedure:

-Create tabs for the following objects:

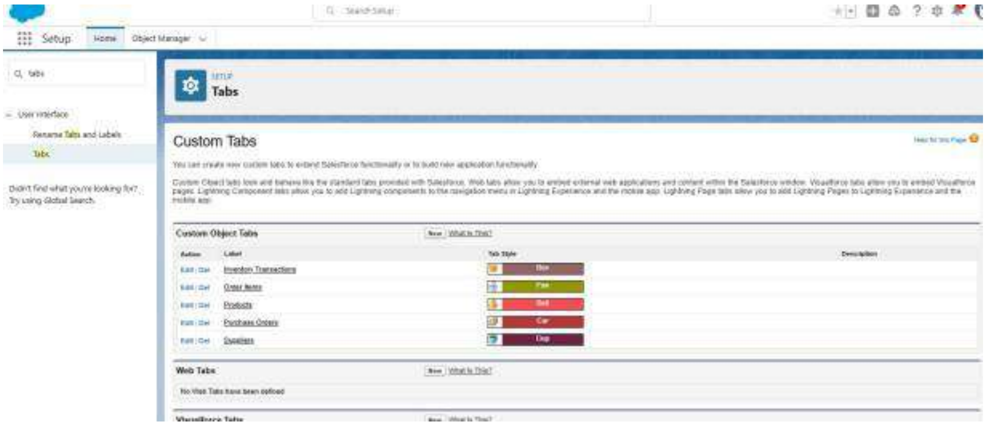
-Purchase Order

-Order Item

-Inventory Transaction

-Supplier

Follow the same steps as described in **Activity 1 (Creating a Tab for the Product Object)** to complete the process for each object.



SETUP
Tabs

Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs

Action	Label	Tab Style	Description
Edit | Del	Inventory Transactions	Box	
Edit | Del	Order Items	Fan	
Edit | Del	Products	Bell	
Edit | Del	Purchase Orders	Car	
Edit | Del	Suppliers	Dog	

Web Tabs

No Web Tabs have been defined

Visualforce Tabs

Milestone 4: The Lightning App

Activity 1: Creating a Lightning App for Medical Inventory Management

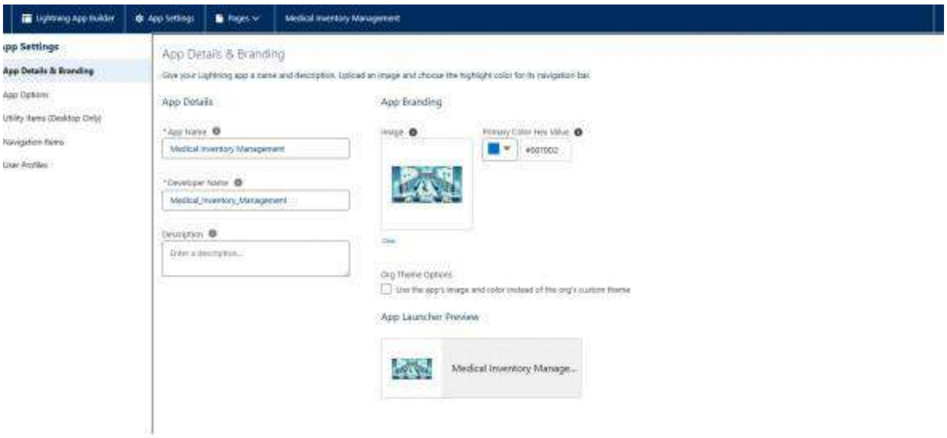
Procedure:

1. From **Setup**, enter **App Manager** in the Quick Find bar and select **App Manager**.
2. Click **New Lightning App**.
3. Enter **Medical Inventory Management** as the **App Name**.

    -Optionally, upload an image related to medical inventory.

    -Click **Next**.

4. Under **App Options**, leave the default selections and click **Next**.
5. Under **Utility Items**, retain the default configuration and click **Next**.



App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name: Medical Inventory Management

*Developer Name: Medical_Inventory_Management

Description: Enter a description...

App Branding

Image

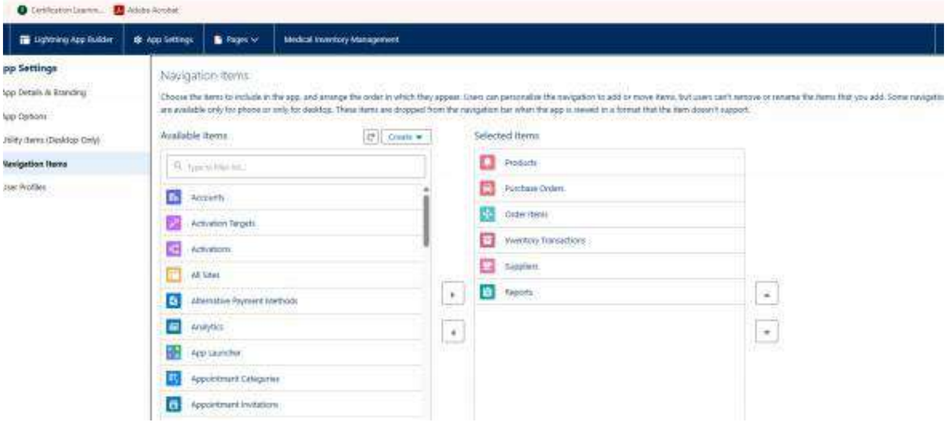
Primary Color Hex Value: #0070D2

Org Theme Options

□ Use the app's image and color instead of the org's custom theme

App Launcher Preview

Medical Inventory Manage...



Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigatio are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items: Accounts, Activation Targets, Activations, All Sites, Alternative Payment Methods, Analytics, App Launcher, Appointment Categories, Appointment Invitations

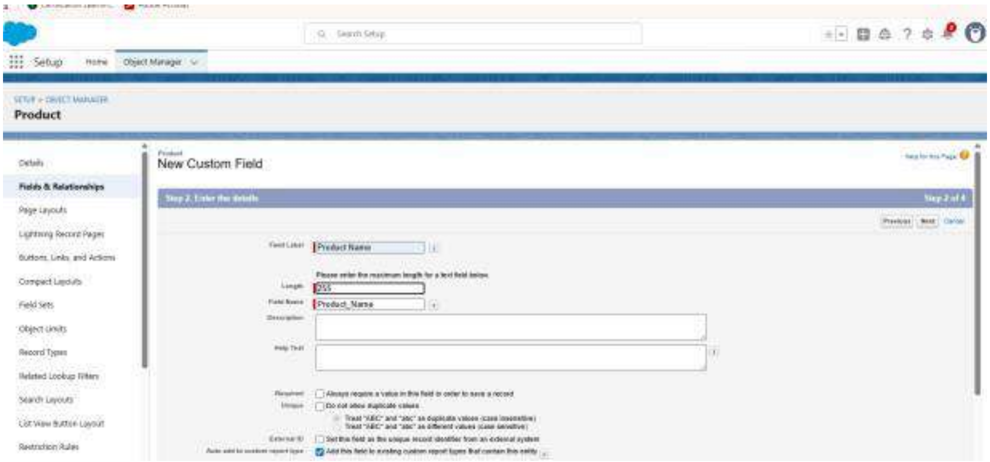
Selected Items: Products, Purchase Orders, Order Items, Inventory Transactions, Suppliers, Reports

Milestone 5: Fields

Activity 1: Creating a Text Field in the Product Object

Steps:

1. Click the **gear icon** and select **Setup** (opens in a new tab).
2. In Setup, go to the **Object Manager** tab.
3. Select the **Product** custom object.
4. From the left navigation, click **Fields & Relationships**.
5. Click **New**.
6. Choose **Text** as the field type and click **Next**.
7. Enter the following details:
   - **Field Label:** Product Name
   - **Length:** 255
8. Select the **Required Field** checkbox.
9. Click **Next** → **Next** → **Save & New** to create the field.



Product
New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: Product Name

Please enter the maximum length for a text field below.

Length: 255

Field Name: Product_Name

Description:

Help Text:

Required: □ Always require a value in this field in order to save a record

Unique: □ Do not allow duplicate values

○ Treat "ABC" and "abc" as duplicate values (case insensitive)

○ Treat "ABC" and "abc" as different values (case sensitive)

External ID: □ Set this field as the unique record identifier from an external system

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Milestone 5: Fields

Activity 2: Creating a Text Area Field in the Product Object

Steps:

-Click the gear icon and select Setup (opens in a new tab).

-In Setup, go to the Object Manager tab.

-Select the Product custom object.

-From the left navigation, click Fields & Relationships.

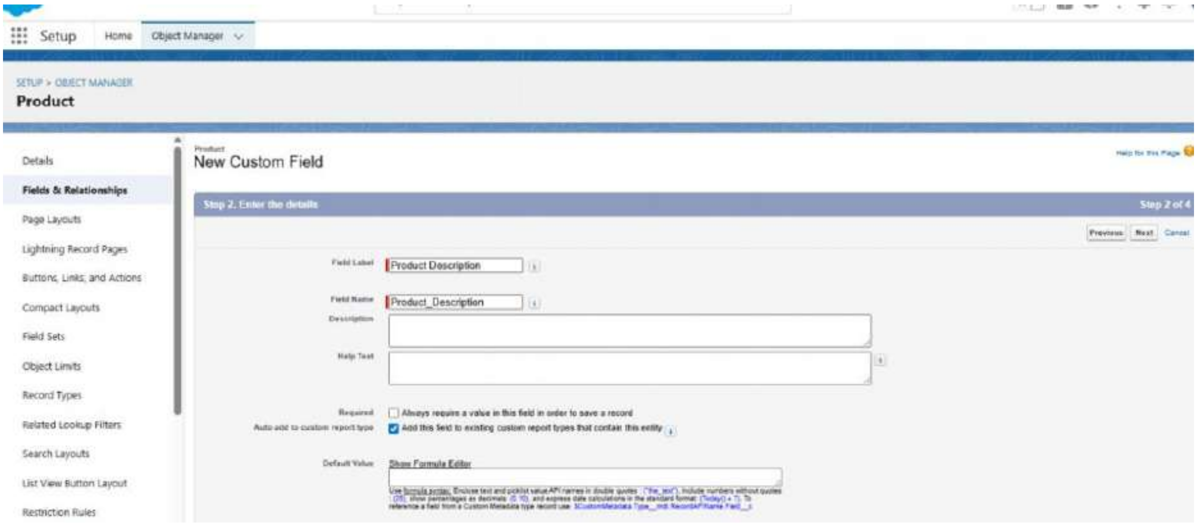
-Click New.

-Choose Text Area as the field type and click Next.

Enter the following details:

Field Label: Product Description

Click Next → Next → Save & New to complete the field creation.



Product
New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: Product Description

Field Name: Product_Description

Description:

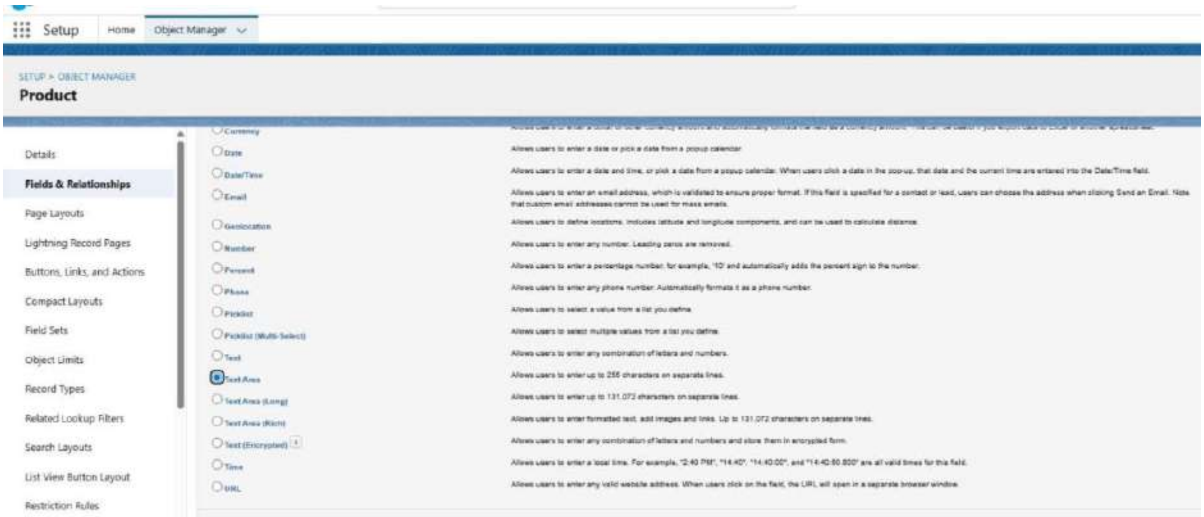
Help Text:

Required: □ Always require a value in this field in order to save a record

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Default Value: Show Formula Editor

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (25), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today() + 7). To reference a field from a Custom Metadata type record use: $CustomMetadata.Type__mdt.RecordAPIName.Field__c



○ Currency	[illegible]
○ Date	Allows users to enter a date or pick a date from a popup calendar.
○ Date/Time	Allows users to enter a date and time, or pick a date from a popup calendar. When users click a date in the pop-up, that date and the current time are entered into the Date/Time field.
○ Email	Allows users to enter an email address, which is validated to ensure proper format. If this field is specified for a contact or lead, users can choose the address when clicking Send an Email. Note that custom email addresses cannot be used for mass emails.
○ Geolocation	Allows users to define locations. Includes latitude and longitude components, and can be used to calculate distance.
○ Number	Allows users to enter any number. Leading zeros are removed.
○ Percent	Allows users to enter a percentage number, for example, '10' and automatically adds the percent sign to the number.
○ Phone	Allows users to enter any phone number. Automatically formats it as a phone number.
○ Picklist	Allows users to select a value from a list you define.
○ Picklist (Multi-Select)	Allows users to select multiple values from a list you define.
○ Text	Allows users to enter any combination of letters and numbers.
● Text Area	Allows users to enter up to 255 characters on separate lines.
○ Text Area (Long)	Allows users to enter up to 131,072 characters on separate lines.
○ Text Area (Rich)	Allows users to enter formatted text, add images and links. Up to 131,072 characters on separate lines.
○ Text (Encrypted)	Allows users to enter any combination of letters and numbers and store them in encrypted form.
○ Time	Allows users to enter a local time. For example, "2:40 PM", "14:40", "14:40:00", and "14:40:50.600" are all valid times for this field.
○ URL	Allows users to enter any valid website address. When users click on the field, the URL will open in a separate browser window.

Activity 3: Creating a Number Field in the Product Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Product and select the Product custom object.

-From the left panel, click Fields & Relationships.

-Click New.

-Choose Number as the data type and click Next.

Enter the details:

Field Label: Current Stock Level

Length: 18

Decimal Places: 0

Click Next → Next → Save to finish creating the field.

Activity 4: Creating a Currency Field in the Product Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Product and select the Product custom object.

-From the left-hand menu, select Fields & Relationships.

-Click New.

-Choose Currency as the data type and click Next.

Enter the details:

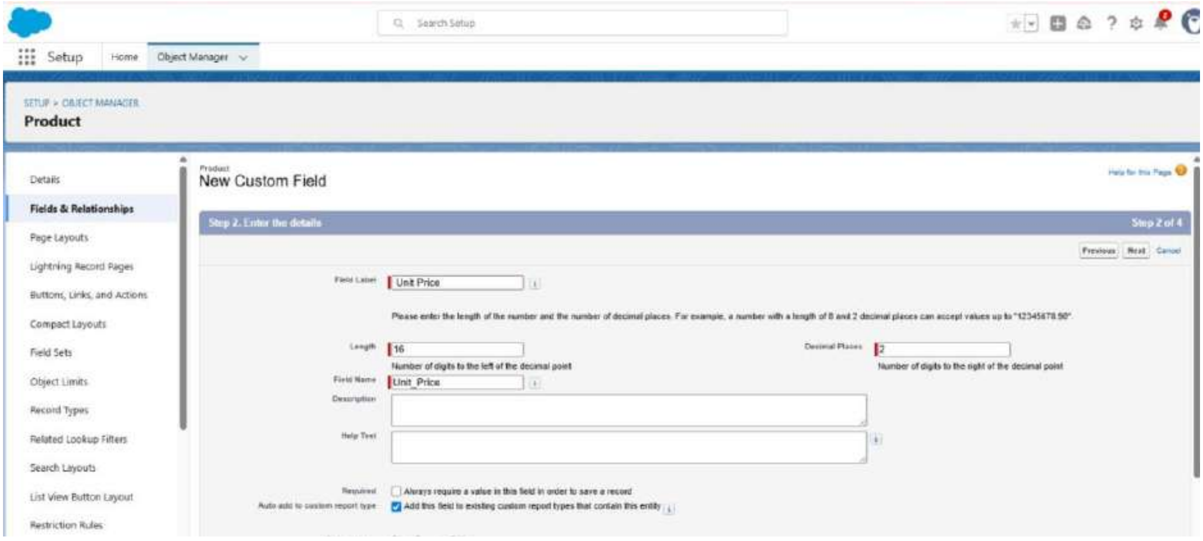
Field Label: Unit Price

Length: 16

Decimal Places: 2

Mark the field as Required.

Click Next → Next → Save.



Product

New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: Unit Price

Please enter the length of the number and the number of decimal places. For example, a number with a length of 8 and 2 decimal places can accept values up to "12345678.90".

Length: 16 — Number of digits to the left of the decimal point

Decimal Places: 2 — Number of digits to the right of the decimal point

Field Name: Unit_Price

Description:

Help Text:

Required: □ Always require a value in this field in order to save a record

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Activity 5: Creating a Lookup Relationship in the Purchase Order Object

A Lookup Relationship in Salesforce links two objects together, where one object (child) references another (parent). This helps maintain relational data integrity and allows easy navigation between related records.

In this activity, we'll establish a relationship from Purchase Order (child) to Supplier (parent).

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Purchase Order and select the Purchase Order custom object.

-From the left-hand menu, click Fields & Relationships.

-Click New.

-Select Lookup Relationship as the data type and click Next.

-For the related object, select Supplier.

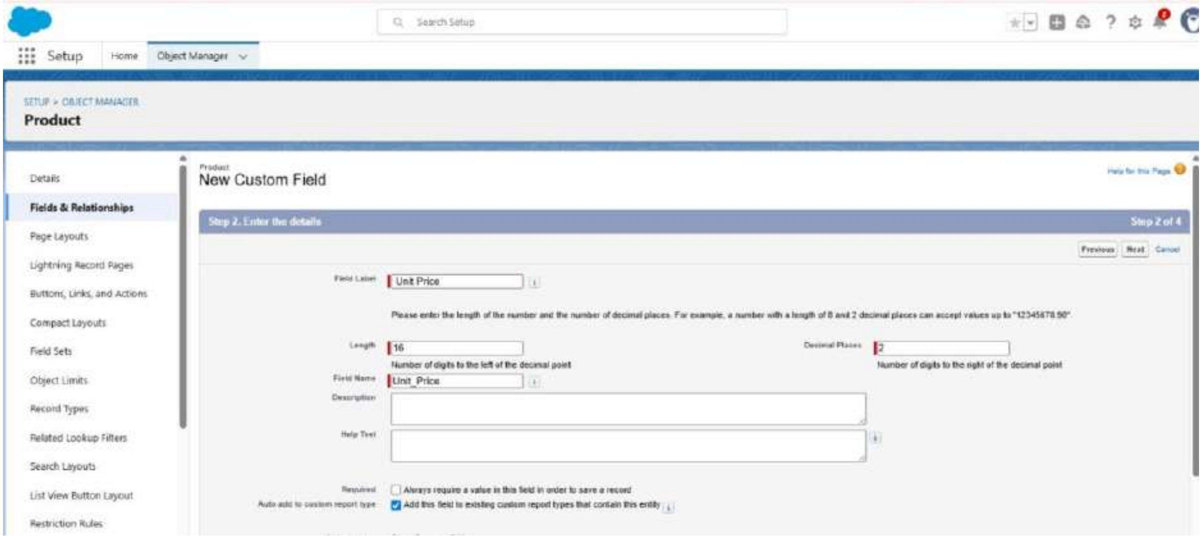
-Click Next.

Enter the details:

Field Label: Supplier ID

Mark the field as Required.

Continue by clicking Next → Next → Next → Save.



Product

New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: Unit Price

Please enter the length of the number and the number of decimal places. For example, a number with a length of 8 and 2 decimal places can accept values up to "12345678.90".

Length: 16 — Number of digits to the left of the decimal point

Decimal Places: 2 — Number of digits to the right of the decimal point

Field Name: Unit_Price

Description:

Help Text:

Required: □ Always require a value in this field in order to save a record

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Activity 6: Creating a Date Field in the Purchase Order Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Purchase Order and select the Purchase Order custom object.

-From the left-hand menu, click Fields & Relationships.

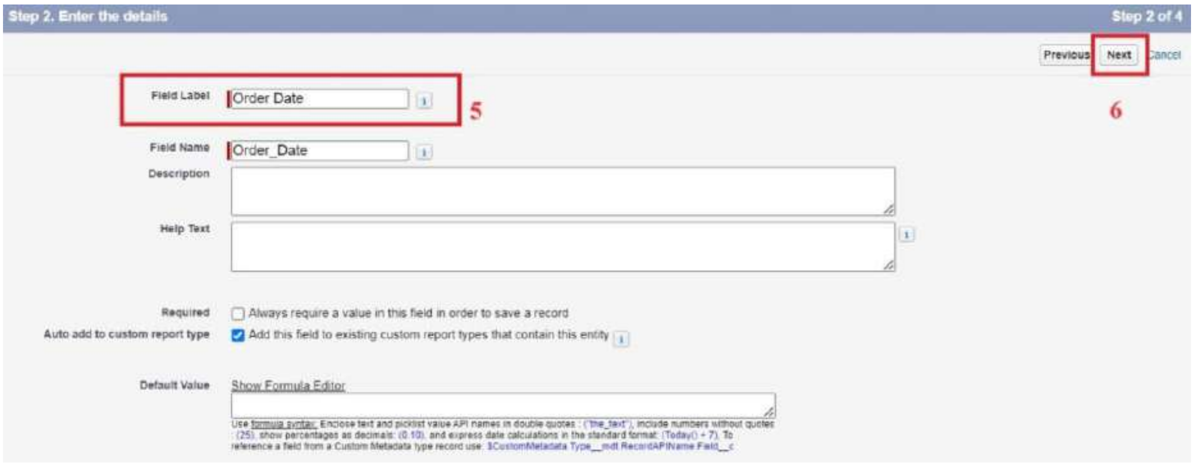
-Click New.

-Select Date as the data type and click Next.

-Enter the following details:

-Field Label: Order Date

-Click Next → Next → Save to complete the creation of the date field.



Step 2. Enter the details — Step 2 of 4

Field Label: Order Date

Field Name: Order_Date

Description:

Help Text:

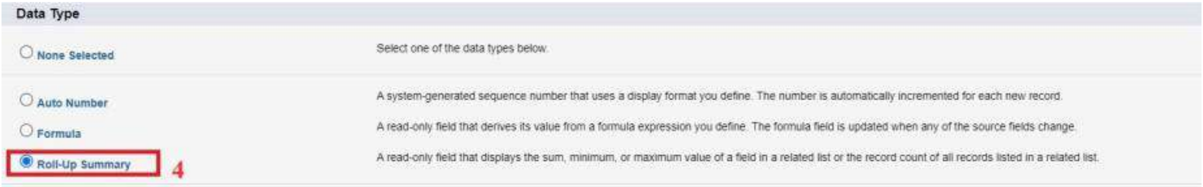
Required: □ Always require a value in this field in order to save a record

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Default Value:

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (25), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today() + 7). To reference a field from a Custom Metadata type record use: $CustomMetadata.Type__mdt.RecordAPIName.Field__c

Activity 7: Creating a Roll-Up Summary Field in Purchase Order object
To create fields in an object:
1. Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box>> click on the Purchase Order object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Roll-Up Summary” and click Next.
5. Enter Field Label as “ Order Count”.
6. Choose the Summarized Object as “Order Items”.
7. For Select Roll-Up Type select “Count”.
8. Click on Next, Next and Save



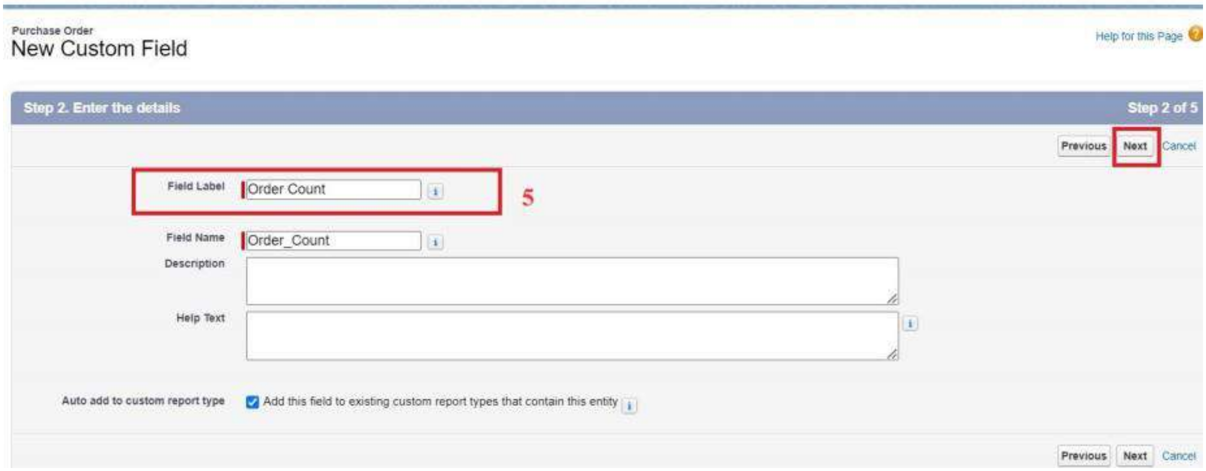
Data Type

○ None Selected — Select one of the data types below.

○ Auto Number — A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.

○ Formula — A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.

● Roll-Up Summary — A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.



Purchase Order
New Custom Field

Step 2. Enter the details — Step 2 of 5

Field Label: Order Count

Field Name: Order_Count

Description:

Help Text:

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Activity 8: Creating a Unit Price Formula Field in the Order Item Object

Steps:

-Go to **Setup** → click on **Object Manager**.

-In the Quick Find box, type **Order Item** and select the **Order Item** custom object.

-From the left-hand menu, click **Fields & Relationships**.

-Click **New**.

-Select **Formula** as the data type and click **Next.**

Enter the following details:

Field Label: Unit Price

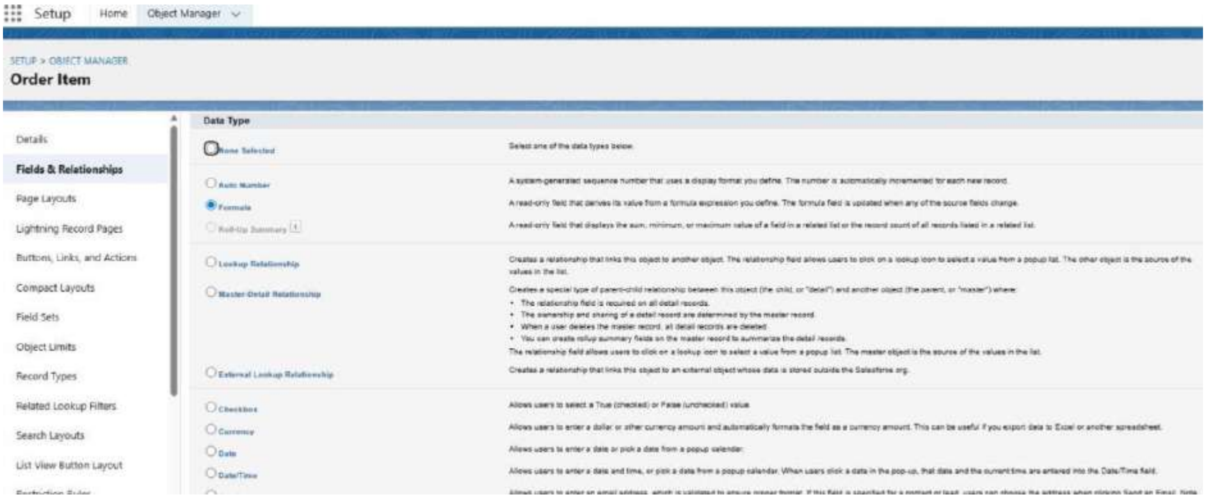
Formula Return Type: Currency

In the formula editor, enter the advanced formula:

Product_ID__r.Unit_Price__c

This pulls the **Unit Price** directly from the related **Product** object.

Click **Next** → **Next** → **Save** to complete the field creation.



Data Type	
○ None Selected	Select one of the data types below.
○ Auto Number	A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.
● Formula	A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.
○ Roll-Up Summary	A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.
○ Lookup Relationship	Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list.
○ Master-Detail Relationship	Creates a special type of parent-child relationship between this object (the child, or "detail") and another object (the parent, or "master") where: • The relationship field is required on all detail records. • The ownership and sharing of a detail record are determined by the master record. • When a user deletes the master record, all detail records are deleted. • You can create rollup summary fields on the master record to summarize the detail records. The relationship field allows users to click on a lookup icon to select a value from a popup list. The master object is the source of the values in the list.
○ External Lookup Relationship	Creates a relationship that links this object to an external object whose data is stored outside the Salesforce org.
○ Checkbox	Allows users to select a True (checked) or False (unchecked) value.
○ Currency	Allows users to enter a dollar or other currency amount and automatically formats the field as a currency amount. This can be useful if you export data to Excel or another spreadsheet.
○ Date	Allows users to enter a date or pick a date from a popup calendar.
○ Date/Time	Allows users to enter a date and time, or pick a date from a popup calendar. When users click a date in the pop-up, that date and the current time are entered into the Date/Time field.

Activity 9: Creating an Amount Formula Field in the Order Item Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Order Item and select the Order Item custom object.

-From the left-hand menu, click Fields & Relationships.

-Click New.

-Select Formula as the data type and click Next.

Enter the following details:

Field Label: Amount

Formula Return Type: Currency

In the formula editor, enter the advanced formula:

Quantity_Received__c * Unit_Price__c

This calculates the total price for each Order Item automatically.

Click Next → Next → Save to complete the field creation.

Activity 10: Creating a Picklist Field in the Inventory Transaction Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Inventory Transaction and select the object.

-From the left-hand menu, click Fields & Relationships.

-Click New.

-Select Picklist as the data type and click Next.

Enter the following details:

Field Label: Transaction Type

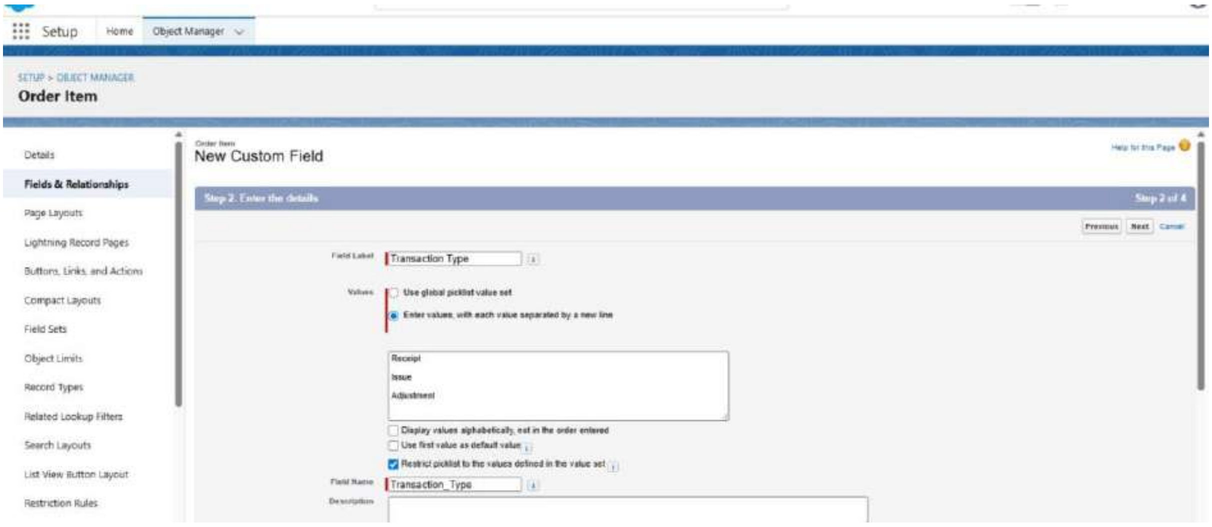
Values: Enter manually, each on a new line:

Receipt

Issue

Adjustment

Click Next → Next → Save to complete the picklist creation.



Order Item
New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: Transaction Type

Values:
○ Use global picklist value set
● Enter values, with each value separated by a new line

Receipt
Issue
Adjustment

□ Display values alphabetically, not in the order entered
□ Use first value as default value
☑ Restrict picklist to the values defined in the value set

Field Name: Transaction_Type

Description:

Activity 11: Creating a Total Order Cost Formula Field in the Inventory Transaction Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Inventory Transaction and select the object.

-From the left-hand menu, click Fields & Relationships.

-Click New.

-Select Formula as the data type and click Next.

-Enter the following details:

Field Label: Total Order Cost

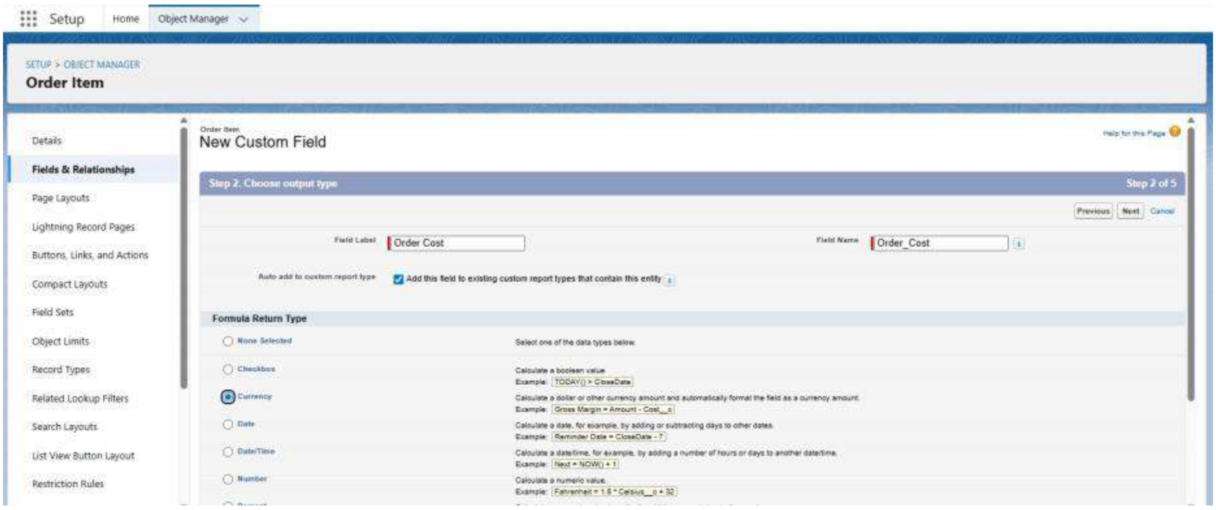
Formula Return Type: Currency

-0In the formula editor, enter the advanced formula:

Purchase_Order_ID__r.Total_Order_Cost__c

-This formula pulls the total cost from the related Purchase Order, ensuring accurate cost tracking for inventory transactions.

-Click Next → Next → Save to complete the field creation.



New Custom Field

Step 2. Choose output type — Step 2 of 5

Field Label: Order Cost

Field Name: Order_Cost

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Formula Return Type

○ None Selected — Select one of the data types below.

○ Checkbox — Calculate a boolean value
Example: TODAY() > CloseDate

● Currency — Calculate a dollar or other currency amount and automatically format the field as a currency amount.
Example: Gross Margin = Amount - Cost__c

○ Date — Calculate a date, for example, by adding or subtracting days to other dates.
Example: Reminder Date = CloseDate - 7

○ Date/Time — Calculate a date/time, for example, by adding a number of hours or days to another date/time.
Example: Next = NOW() + 1

○ Number — Calculate a numeric value.
Example: Fahrenheit = 1.8 * Celsius__c + 32

Activity 12: Creating a Phone Field in the Supplier Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Supplier and select the Supplier custom object.

-From the left-hand menu, click Fields & Relationships.

-Click New.

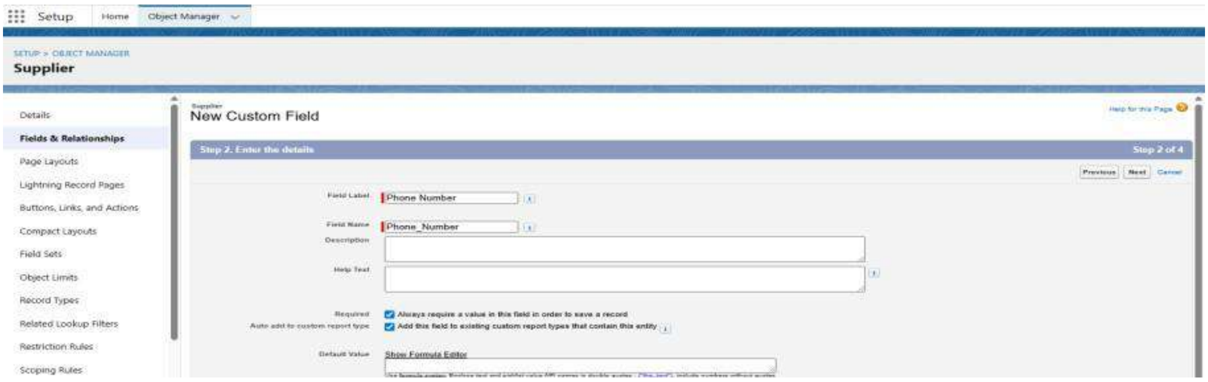
-Select Phone as the data type and click Next.

Enter the following details:

Field Label: Phone Number

-Mark the field as Required.

-Click Next → Next → Save to complete the field creation.



New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: Phone Number

Field Name: Phone_Number

Description:

Help Text:

Required: ☑ Always require a value in this field in order to save a record

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Default Value:

Activity 13: Creating an Email Field in the Supplier Object

Steps:

Go to **Setup** → click on **Object Manager**.

In the Quick Find box, type **Supplier** and select the **Supplier** custom object.

From the left-hand menu, click **Fields & Relationships**.

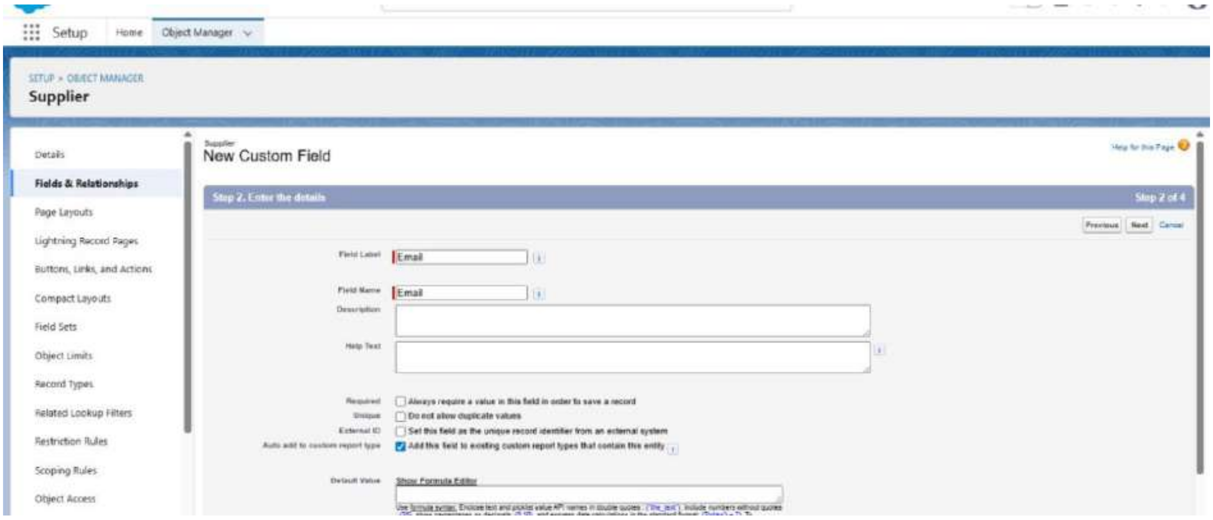
Click **New**.

Select **Email** as the data type and click **Next**.

Enter the following details:

Field Label: Email

Click **Next** → **Next** → **Save** to complete the field creation.



Supplier

New Custom Field

Step 2. Enter the details

Step 2 of 4

Field Label: Email

Field Name: Email

Description:

Help Text:

Required: □ Always require a value in this field in order to save a record

Unique: □ Do not allow duplicate values

External ID: □ Set this field as the unique record identifier from an external system

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Default Value:

Milestone 6: Page Layout Customization

Activity 1: Editing a Page Layout in the Product Object

Steps:

-Go to Setup → click on Object Manager.

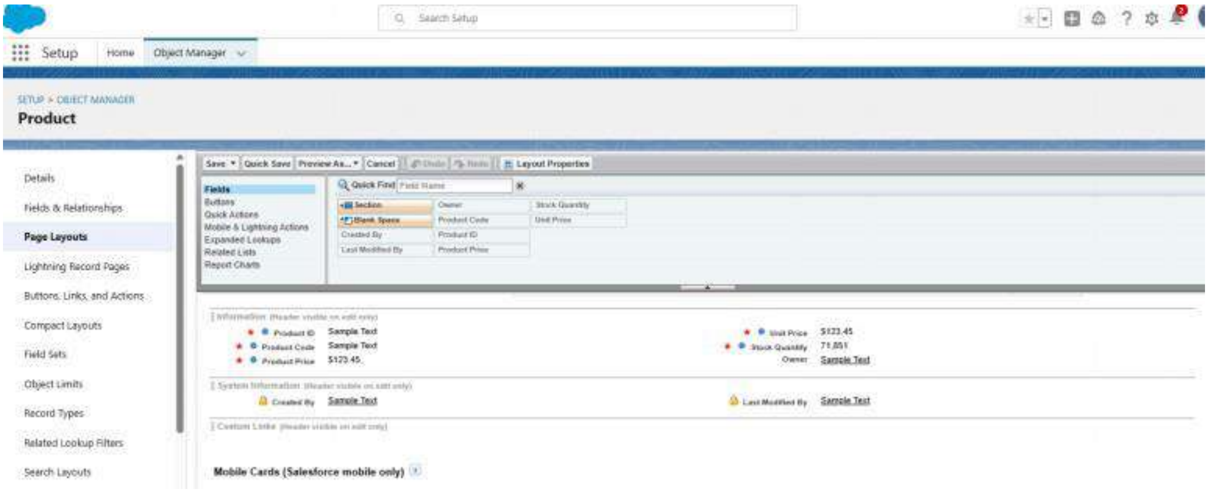
-In the Quick Find box, type Product and select the Product custom object.

-From the left-hand menu, click Page Layouts.

-Select the layout named Product Layout.

-Drag and arrange the fields on the page layout as required to optimize data entry and display.

- Save it



Information (Header visible on edit only)

Product ID Sample Text

Product Code Sample Text

Product Price $123.45

Unit Price $123.45

Stock Quantity 71,851

Owner Sample Text

System Information (Header visible on edit only)

Created By Sample Text

Last Modified By Sample Text

Custom Links (Header visible on edit only)

Mobile Cards (Salesforce mobile only)

Activity 2: Editing a Page Layout in the Purchase Order Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Purchase Order and select the Purchase Order custom object.

-From the left-hand menu, click Page Layouts.

-Select the layout named Purchase Order Layout.

-Drag and arrange the fields on the layout as required to optimize data entry and display.

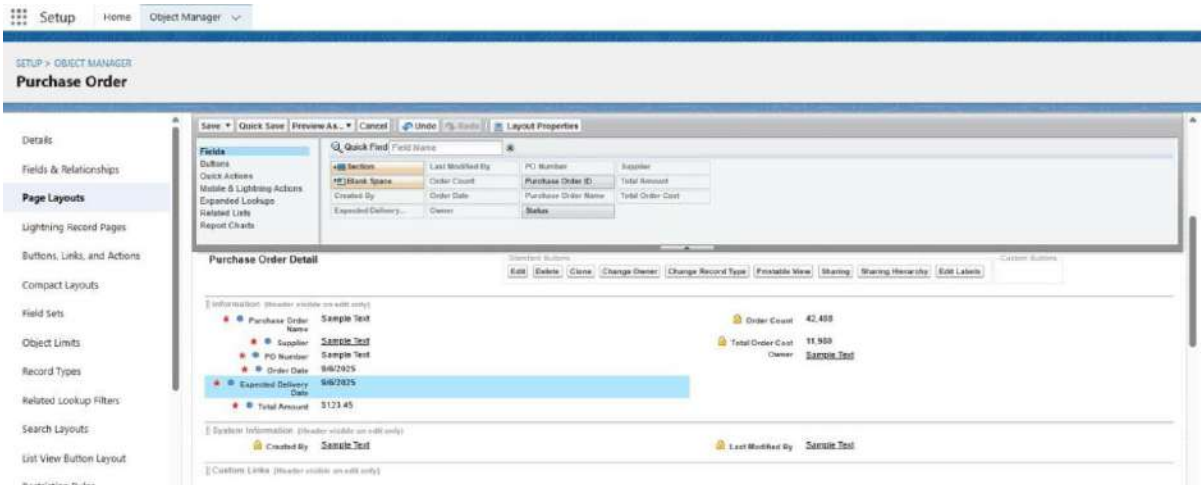
For the Order Date **field:**

-Click on the field → click Settings → select Required → save.

-For the Total Order Cost field:

-Click on the field → click Settings → select Read-Only → save.

-Click Save to finalize the layout changes.



Information (Header visible on edit only)

Purchase Order Name Sample Text

Supplier Sample Text

PO Number Sample Text

Order Date 9/6/2025

Expected Delivery Date 9/6/2025

Total Amount $123.45

Order Count 42,488

Total Order Cost 11,980

Owner Sample Text

System Information (Header visible on edit only)

Created By Sample Text

Last Modified By Sample Text

Custom Links (Header visible on edit only)

Activity 3: Editing a Page Layout in the Order Item Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Order Item and select the Order Item custom object.

-From the left-hand menu, click Page Layouts.

-Select the layout named Order Item Layout.

-Drag and arrange the fields on the layout as required to optimize data entry and display.

-Click Save to finalize the layout changes.

Activity 4: Editing a Page Layout in the Inventory Transaction Object

Steps:

-Go to Setup → click on Object Manager.

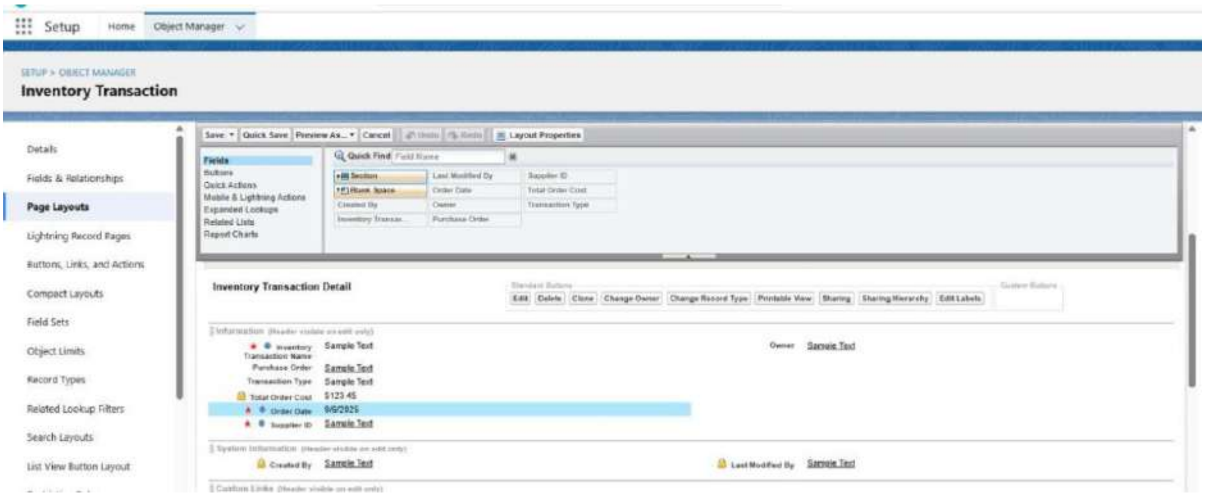
-In the Quick Find box, type Inventory Transaction and select the Inventory Transaction custom object.

-From the left-hand menu, click Page Layouts.

-Select the layout named Inventory Transaction Layout.

-Drag and arrange the fields on the layout as required to optimize data entry and display.

-Click Save to finalize the layout changes.



Inventory Transaction Name: Sample Text

Owner: Sample Text

Purchase Order: Sample Text

Transaction Type: Sample Text

Total Order Cost: $123.45

Order Date: 9/6/2025

Supplier ID: Sample Text

Created By: Sample Text

Last Modified By: Sample Text

Activity 5: Editing a Page Layout in the Supplier Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Supplier and select the Supplier custom object.

-From the left-hand menu, click Page Layouts.

--Select the layout named Supplier Layout.

-Drag and arrange the fields on the layout as required to optimize data entry and display.

-Click Save to finalize the layout changes.

Milestone 7: Compact Layouts

Activity 1: Creating a Compact Layout for the Product Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Product and select the Product custom object.

-From the sidebar, click Compact Layouts.

-Click New.

-Enter the following details:

Label: Product Compact Layout

-Select the fields to display in the compact layout:

Product Name

Unit Price

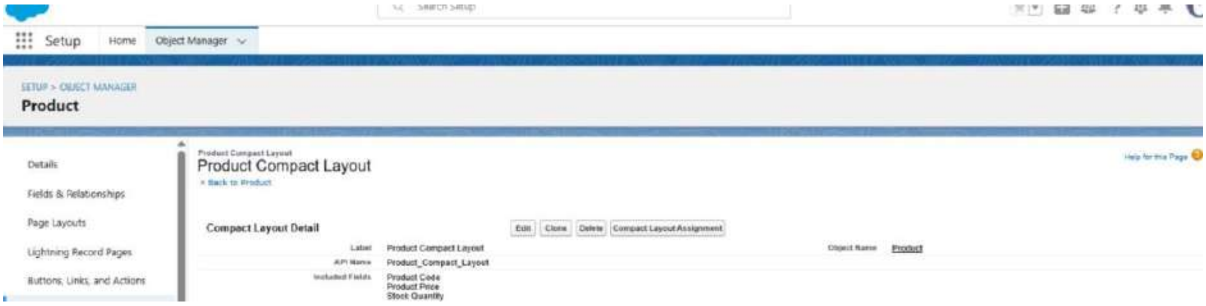
Current Stock Level

-Click Save.

-Click Compact Layout Assignment.

-Click Edit Assignment.

-Choose Product Compact Layout from the dropdown and click Save.



Product

Product Compact Layout

Product Compact Layout

Compact Layout Detail

Label: Product Compact Layout

API Name: Product_Compact_Layout

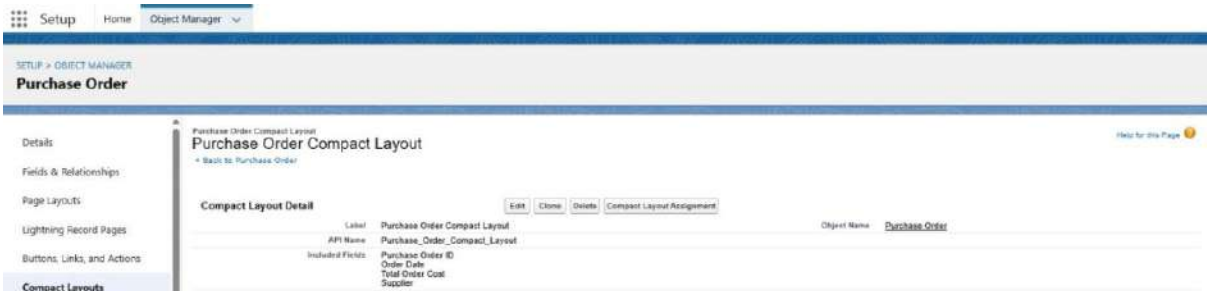
Included Fields: Product Code
Product Price
Stock Quantity

Object Name: Product

Activity 2: Creating a Compact Layout for the Purchase Order Object

Steps:

1. Go to **Setup** → click on **Object Manager**.
2. In the Quick Find box, type **Purchase Order** and select the **Purchase Order** custom object.
3. From the sidebar, click **Compact Layouts**.
4. Click **New**.
5. Enter the following details:
   - **Label:** Purchase Order Compact Layout
6. Select the fields to display in the compact layout:
   - Purchase Order ID
   - Order Date
   - Total Order Cost
   - Supplier ID
7. Click **Save**.
8. Click **Compact Layout Assignment** → **Edit Assignment**.
9. Choose **Purchase Order Compact Layout** from the dropdown.
10. Click **Save**.



Purchase Order

Purchase Order Compact Layout

Purchase Order Compact Layout

Compact Layout Detail

Label: Purchase Order Compact Layout

API Name: Purchase_Order_Compact_Layout

Included Fields: Purchase Order ID
Order Date
Total Order Cost
Supplier

Object Name: Purchase Order

Milestone 8: Validation Rules

Activity 1: Creating an Expected Delivery Date Validation Rule for the Purchase Order Object

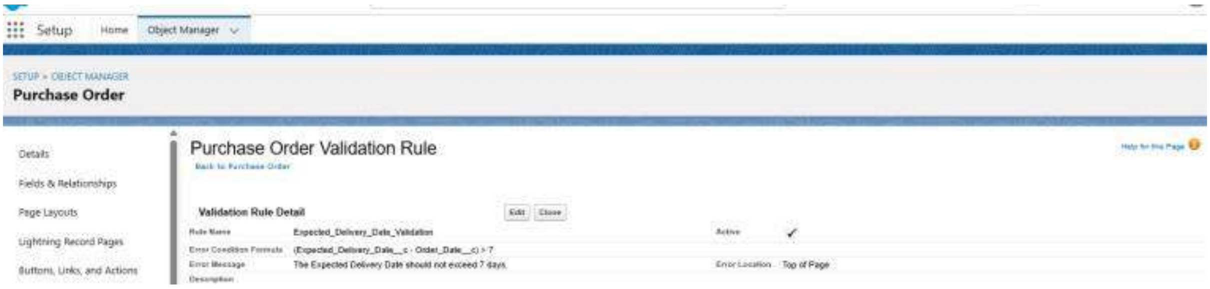
Steps:

1. Go to **Setup** → click on **Object Manager**.
2. In the Quick Find box, type **Purchase Order** and select the **Purchase Order** custom object.

3. From the left-hand menu, click **Validation Rules → New**.
4. Enter the following details:
   - **Rule Name:** Expected Delivery Date Validation
   - **Active:** Checked
5. In the formula editor, enter the error condition formula:
6. (Expected_Delivery_Date__c - Order_Date _ c) > 7

   This ensures that the expected delivery date cannot exceed 7 days from the order date.

7. Click **Save** to activate the validation rule.



Purchase Order Validation Rule

Back to Purchase Order

Validation Rule Detail

Rule Name	Expected_Delivery_Date_Validation	Active	✓
Error Condition Formula	(Expected_Delivery_Date__c - Order_Date__c) > 7		
Error Message	The Expected Delivery Date should not exceed 7 days.	Error Location	Top of Page
Description			

Milestone 9: Profiles

Activity 1: Creating an Inventory Manager Profile

Steps:

Go to Setup → type Profiles in the Quick Find box → click Profiles.

Locate Standard User → click Clone.

Enter the Profile Name: Inventory Manager → click Save.

On the newly created profile page, click Edit.

Configure the following settings:

Custom App Settings: Set Medical Inventory Management as default.

Password Policies:

User passwords expire in: Never Expires

Minimum password length: 8

Click Save.

Activity 2: Creating a Purchase Manager Profile

Steps:

Go to Setup → type Profiles in the Quick Find box → click Profiles.

Locate Standard User → click Clone.

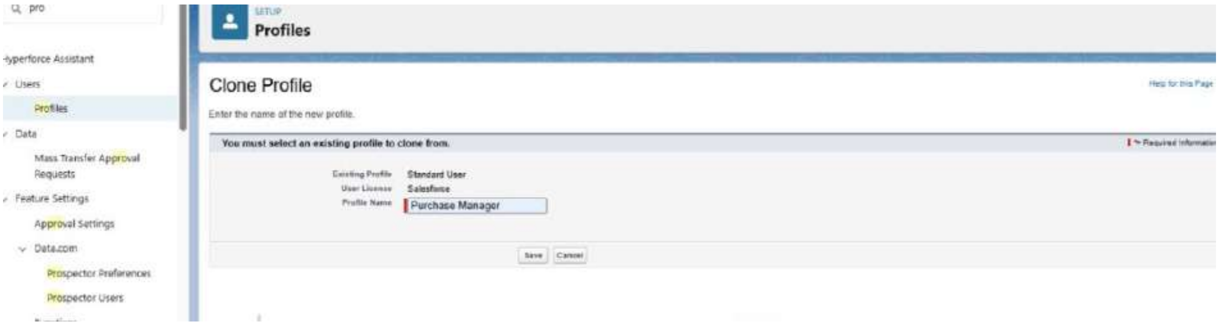
Enter the Profile Name: Purchase Manager → click Save.

On the newly created profile page, click Edit.

Configure the following settings:

Custom App Settings: Set Medical Inventory Management as default.

Click Save.



Clone Profile

Enter the name of the new profile.

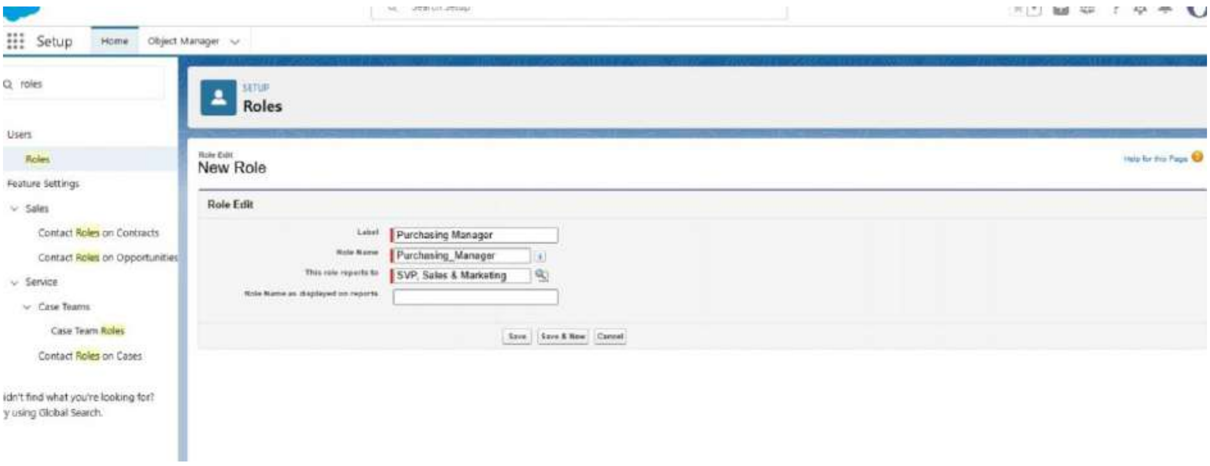
You must select an existing profile to clone from.

Existing Profile: Standard User
User License: Salesforce
Profile Name: Purchase Manager

Milestone 10: Roles
Activity 1: Creating a Purchasing Manager Role

Steps:

1. Go to Setup → type Roles in the Quick Find box → click Set Up Roles.
2. Click Expand All to view the role hierarchy.
3. Under the SVP, Sales & Marketing role, click Add Role.
4. Enter the following details:
   - Label: Purchasing Manager
   - The Role Name will auto-populate.
5. Click Save to create the role.



Role Edit
New Role

Role Edit

Label: Purchasing Manager
Role Name: Purchasing_Manager
This role reports to: SVP, Sales & Marketing
Role Name as displayed on reports:

Activity 2: Creating an Inventory Manager Role

Steps:

Go to Setup → type Roles in the Quick Find box → click Set Up Roles.

Click Expand All to view the role hierarchy.

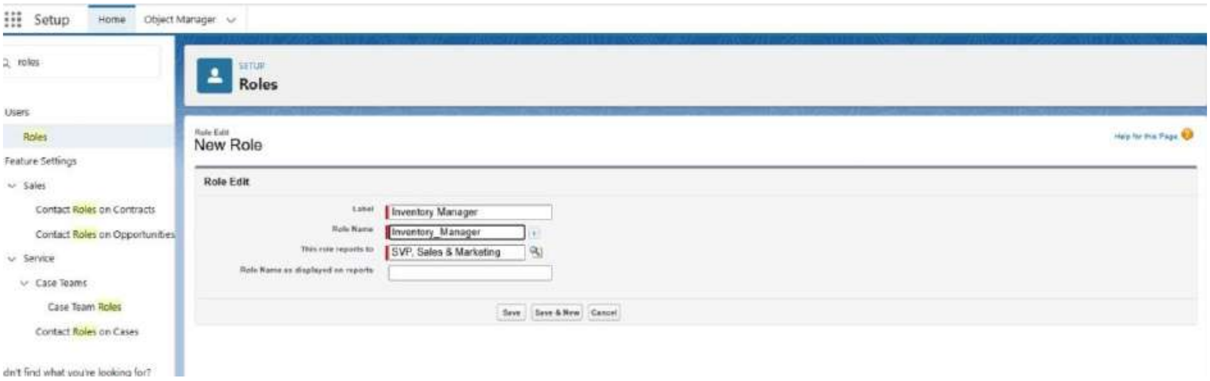
Under the SVP, Sales & Marketing role, click Add Role.

Enter the following details:

Label: Inventory Manager

The Role Name will auto-populate.

Click Save to create the role.



Role Edit

New Role

Role Edit

Label: Inventory Manager

Role Name: Inventory_Manager

This role reports to: SVP, Sales & Marketing

Role Name as displayed on reports:

Milestone 12: Permission Sets

Activity 1: Creating a Permission Set

Steps:

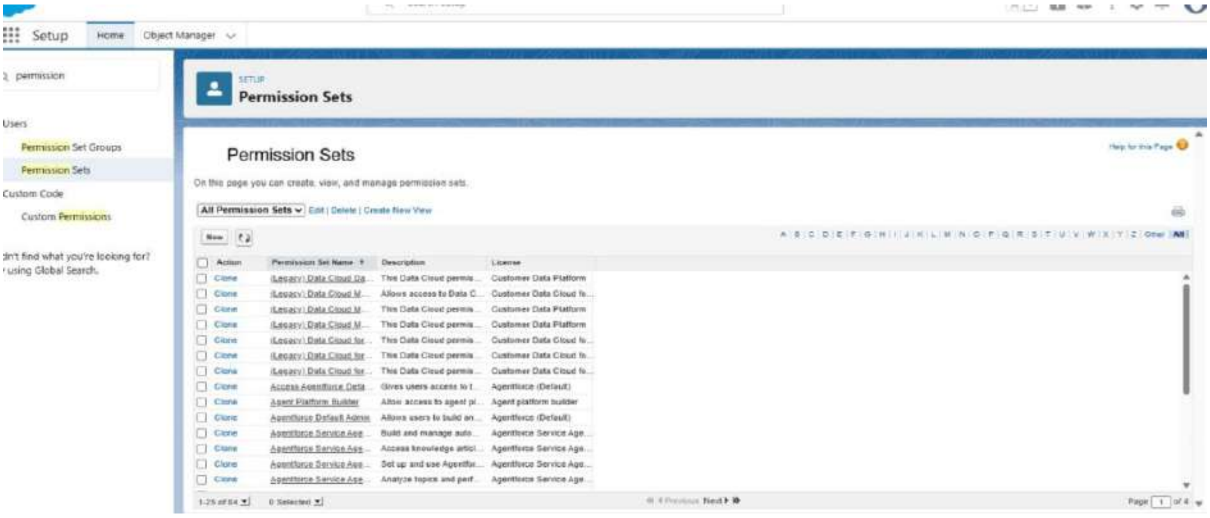
Go to Setup → type Permission in the Quick Find box → select Permission Sets.

Click New.

Enter the following details:

Label: Purchase Manager Create Access

Click Save to create the permission set.



Permission Sets

On this page you can create, view, and manage permission sets.

All Permission Sets | Edit | Delete | Create New View

□ Action	Permission Set Name ↑	Description	License
□ Clone	(Legacy) Data Cloud Da...	The Data Cloud permis...	Customer Data Platform
□ Clone	(Legacy) Data Cloud M...	Allows access to Data C...	Customer Data Cloud fo...
□ Clone	(Legacy) Data Cloud M...	This Data Cloud permis...	Customer Data Platform
□ Clone	(Legacy) Data Cloud M...	This Data Cloud permis...	Customer Data Platform
□ Clone	(Legacy) Data Cloud for...	This Data Cloud permis...	Customer Data Cloud fo...
□ Clone	(Legacy) Data Cloud for...	The Data Cloud permis...	Customer Data Cloud fo...
□ Clone	(Legacy) Data Cloud for...	This Data Cloud permis...	Customer Data Cloud fo...
□ Clone	Access Agentforce Data...	Gives users access to t...	Agentforce (Default)
□ Clone	Agent Platform Builder	Allow access to agent pl...	Agent platform builder
□ Clone	Agentforce Default Admin	Allows users to build an...	Agentforce (Default)
□ Clone	Agentforce Service Age...	Build and manage auto...	Agentforce Service Age...
□ Clone	Agentforce Service Age...	Access knowledge articl...	Agentforce Service Age...
□ Clone	Agentforce Service Age...	Set up and use Agentfor...	Agentforce Service Age...
□ Clone	Agentforce Service Age...	Analyze topics and perf...	Agentforce Service Age...

1-25 of 64 | 0 Selected | Previous Next | Page 1 of 4

Milestone 13: Flows

Activity 1: Creating a Flow to Update the Actual Delivery Date

Steps:

Go to Setup → type Flow in the Quick Find box → click Flows → New Flow → select Start From Scratch.

Choose Record-Triggered Flow → click Create.

Under Object, select Purchase Order.

Configure the trigger: A record is created or updated.

Set Entry Conditions: None.

Select Fast Field Updates → click Done.

Get Records Element
7. Click the "+" icon → select Get Records.
8. Enter Label: Get Purchase Record.
9. Select Object: Purchase Order.
10. For Condition Requirements, choose All Conditions Are Met (AND).
11. Set the condition:

Field: Id

Operator: Equals

Value: {!$Record.Id}

How Many Records to Store: Only the First Record.

How to Store Record Data: Choose fields and let Salesforce do the rest → select Order_Date__c → click Done.

Create a Variable
14. In Flow Builder, click Manager → New Resource.
15. Resource Type: Variable
16. API Name: ActualDeliveryDate
17. Data Type: Date → click Done.

Assignment Element
18. Drag and drop Assignment from the Toolbox.
19. Enter Label: Assignment.
20. Set Variable Values:

Variable: {!ActualDeliveryDate}

Operator: Equals

Value: {!$Record.Order_Date__c}

Variable: {!ActualDeliveryDate}

Operator: Add

Value: 3

Click Done.

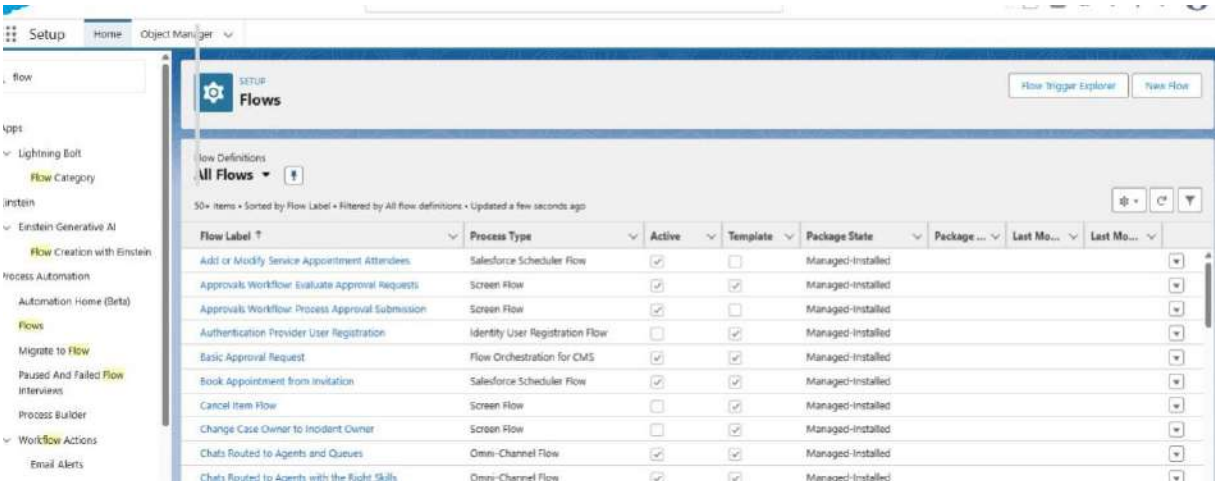
Update Records Element
22. Drag and drop Update Records → connect it to the Assignment element.
23. Enter Label: Updating Purchase Order.
24. How to Find Records to Update: Use the Purchase Order record that triggered the flow.
25. Filter Conditions: None – Always Update Record.
26. Set Field Values:

Field: Actual_Delivery_Date__c

Value: {!ActualDeliveryDate}

Click Done.

Save and Activate Flow
28. Save the flow as Actual Delivery Date Updating.
29. Activate the flow.



50+ items • Sorted by Flow Label • Filtered by All flow definitions • Updated a few seconds ago

Flow Label	Process Type	Active	Template	Package State	Package ...	Last Mo...	Last Mo...
Add or Modify Service Appointment Attendees	Salesforce Scheduler Flow	☑	□	Managed-Installed			
Approvals Workflow: Evaluate Approval Requests	Screen Flow	☑	☑	Managed-Installed			
Approvals Workflow: Process Approval Submission	Screen Flow	☑	□	Managed-Installed			
Authentication Provider User Registration	Identity User Registration Flow	□	☑	Managed-Installed			
Basic Approval Request	Flow Orchestration for CMS	☑	☑	Managed-Installed			
Book Appointment from Invitation	Salesforce Scheduler Flow	☑	☑	Managed-Installed			
Cancel Item Flow	Screen Flow	□	☑	Managed-Installed			
Change Case Owner to Incident Owner	Screen Flow	□	☑	Managed-Installed			
Chats Routed to Agents and Queues	Omni-Channel Flow	☑	☑	Managed-Installed			
Chats Routed to Agents with the Right Skills	Omni-Channel Flow	☑	☑	Managed-Installed			

Milestone 14: Triggers

Activity 1: Creating a Trigger to Calculate Total Amount on Order Item

Step 1: Login to Salesforce

Log in to your Salesforce account with administrative privileges.

Step 2: Navigate to Developer Console

Click the **gear icon** (Setup) at the top-right corner → open the **Setup menu**.

Click **Developer Console** → opens in a new browser tab/window.

Step 3: Create the Apex Trigger

In Developer Console, go to **File** → **New** → **Apex Trigger**.

Name the trigger: CalculateTotalAmountTrigger.

Paste the following code:

```
trigger CalculateTotalAmountTrigger on Order_Item__c (after insert, after update,
after delete, after undelete) {
    // Call the handler class to handle the logic
    CalculateTotalAmountHandler.calculateTotal(Trigger.new, Trigger.old,
Trigger.isInsert, Trigger.isUpdate, Trigger.isDelete, Trigger.isUndelete);
}
```

Step 4: Create the Apex Handler Class

In Developer Console, go to **File** → **New** → **Apex Class**.

Name it CalculateTotalAmountHandler.

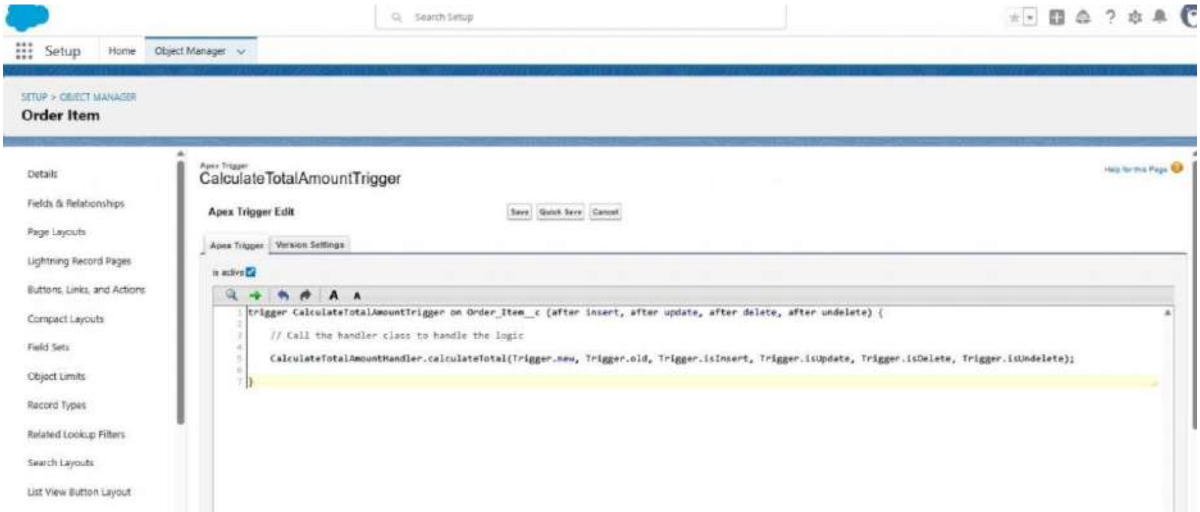
Paste the following code:

```
public class CalculateTotalAmountHandler {
    // Method to calculate the total amount for Purchase Orders based on related
Order Items
    public static void calculateTotal(List<Order_Item__c> newItems,
List<Order_Item__c> oldItems, Boolean isInsert, Boolean isUpdate, Boolean
isDelete, Boolean isUndelete) {

        // Collect Purchase Order IDs affected by changes in Order_Item__c
        Set<Id> parentIds = new Set<Id>();

        // For insert, update, and undelete scenarios
        if (isInsert || isUpdate || isUndelete) {
            for (Order_Item__c ordItem : newItems) {
                parentIds.add(ordItem.Purchase_Order_Id___c);
            }
        }

        // For update and delete scenarios
        if (isUpdate || isDelete) {
            for (Order_Item__c ordItem : oldItems) {
                parentIds.add(ordItem.Purchase_Order_Id___c);
            }
        }

        // Calculate the total amounts for affected Purchase Orders
        Map<Id, Decimal> purchaseToUpdateMap = new Map<Id, Decimal>();
        if (!parentIds.isEmpty()) {
            List<AggregateResult> aggrList = [
                SELECT Purchase_Order_Id__c, SUM(Amount___c) totalAmount
                FROM Order_Item_c
                WHERE Purchase_Order_Id__c IN :parentIds
                GROUP BY Purchase_Order_Id__c
            ];

            for (AggregateResult aggr : aggrList) {
                Id purchaseOrderId = (Id) aggr.get('Purchase_Order_Id__c');
                Decimal totalAmount = (Decimal) aggr.get('totalAmount');
                purchaseToUpdateMap.put(purchaseOrderId, totalAmount);
            }

            // Prepare Purchase Order records for update
            List<Purchase_Order__c> purchaseToUpdate = new
List<Purchase_Order__c>();
            for (Id purchaseOrderId : purchaseToUpdateMap.keySet()) {
                Purchase_Order__c purchaseOrder = new Purchase_Order__c(
                    Id = purchaseOrderId,
                    Total_Order_cost__c = purchaseToUpdateMap.get(purchaseOrderId)
                );
                purchaseToUpdate.add(purchaseOrder);
            }

            // Update Purchase Orders if there are any changes
            if (!purchaseToUpdate.isEmpty()) {
                update purchaseToUpdate;
            }
        }
    }
}
```

Step 5: Save and Test

Click **Save** for both the Trigger and the Handler Class.

Test by creating, updating, or deleting **Order Items**. The **Total Order Cost** on the related Purchase Order should update automatically.



```
trigger CalculateTotalAmountTrigger on Order_Item__c (after insert, after update, after delete, after undelete) {

    // Call the handler class to handle the logic

    CalculateTotalAmountHandler.calculateTotal(Trigger.new, Trigger.old, Trigger.isInsert, Trigger.isUpdate, Trigger.isDelete, Trigger.isUndelete);

}
```

Milestone 15 - Reports

Activity 1: Create a Purchase Orders based on Suppliers(Summary) Report

1. Click App Launcher
2. Select Medical Inventory Management App
3. Click on Reports tab
4. Click on New Report.
5. Click the report type as Purchase Orders Click Start report.
6. Click on Filters and select as follows and click on Apply
7. Customize your report, in group rows select – Supplier ID, Purchase Order: Purchase Order ID, for columns Order Count, Total Order Cost (In this way we are making a Summary Report).
8. Click save and run
9. Give report name – Purchase Orders based on Suppliers.
10. Click Save

NOTE: In this report you can see your all record of the object you selected for reporting
What you selects in "Select a report type option")
(View Report

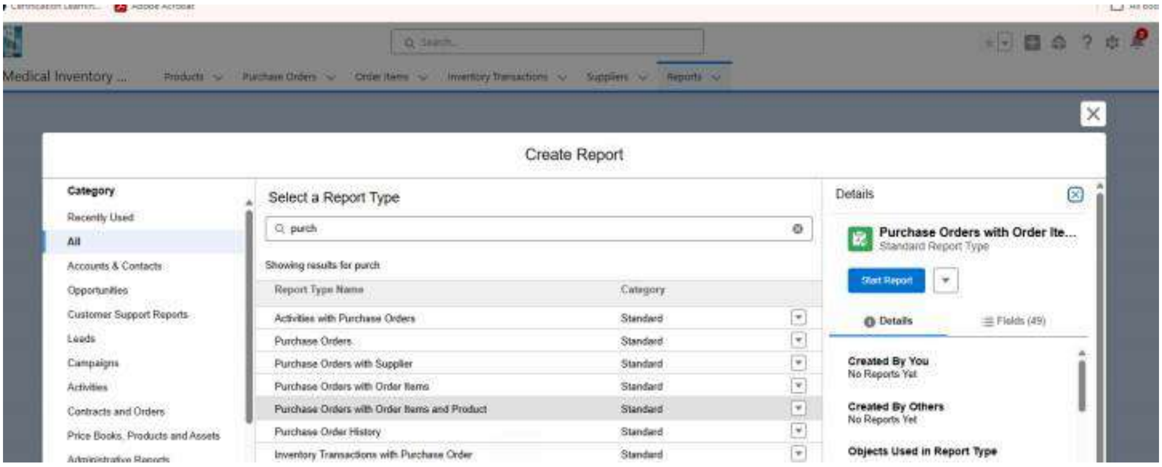
1. Click on App Launcher on the left side of the screen.
2. Search Medical Inventory Management App & click on it.
3. Click on Reports Tab.
4. Click on Purchase Orders based on Suppliers and see records.

Activity 2: Create a Complete Purchase Details Report

1. Click App Launcher
2. Select Medical Inventory Management App
3. Click on Reports tab
4. Click on New Report.
5. Click the report type as Purchase Orders with Order Items and Product ID

>> Click Start report.

6. Click on Filters and select as follows and click on Apply



purch

Showing results for purch

Report Type Name	Category
Activities with Purchase Orders	Standard
Purchase Orders	Standard
Purchase Orders with Supplier	Standard
Purchase Orders with Order Items	Standard
Purchase Orders with Order Items and Product	Standard
Purchase Order History	Standard
Inventory Transactions with Purchase Order	Standard

Purchase Orders with Order Ite...
Standard Report Type

Created By You
No Reports Yet

Created By Others
No Reports Yet

Objects Used in Report Type

Milestone 16: Dashboards

Activity 1: Create Dashboard

Open the Dashboards tab within the Medical Inventory Management application.

Click New Dashboard.

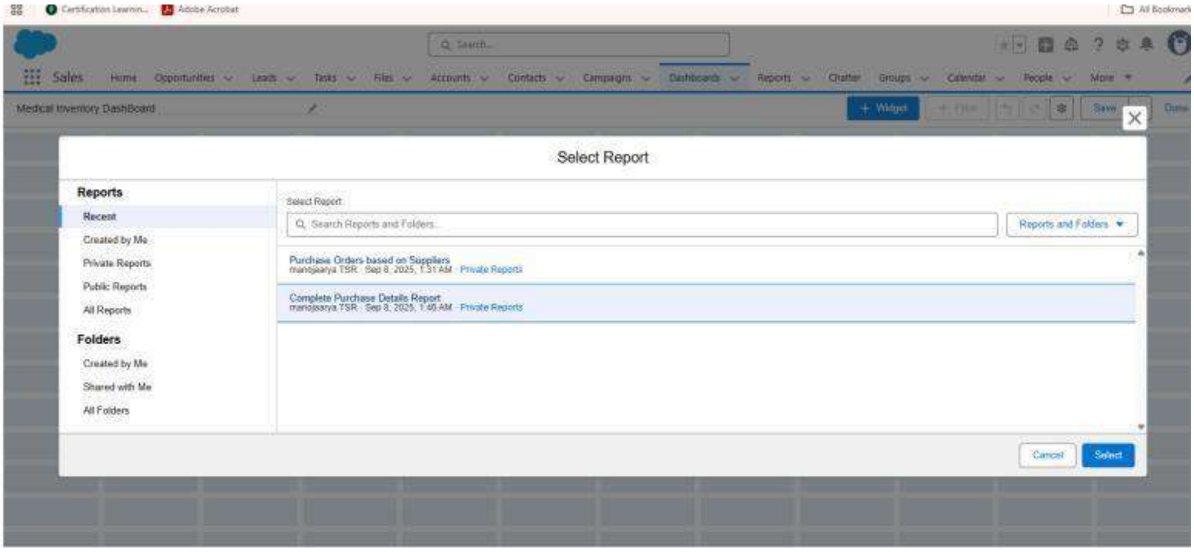
Enter the Name: Medical Inventory Dashboard → Click Create.

Click +Widget to add a component.

Select the Purchase Orders based on Suppliers report.

Choose a suitable data visualization type (chart, table, etc.) based on your requirement.

Click Add → then Save.



Medical Inventory DashBoard

Select Report

Reports

- Recent
- Created by Me
- Private Reports
- Public Reports
- All Reports

Folders

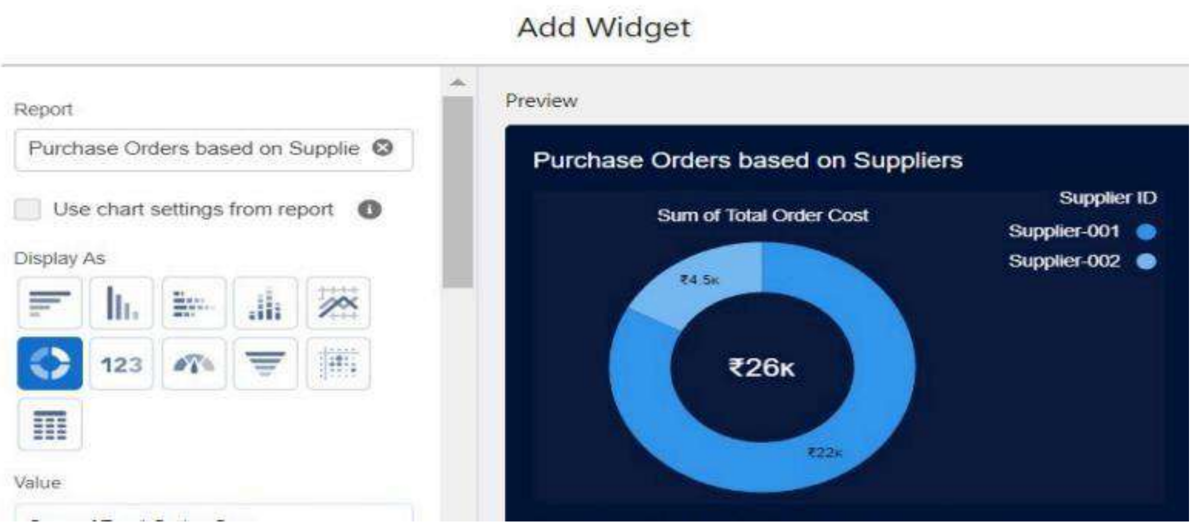
- Created by Me
- Shared with Me
- All Folders

Select Report

Search Reports and Folders...

Purchase Orders based on Suppliers
manojaarya TSR · Sep 8, 2025, 1:31 AM · Private Reports

Complete Purchase Details Report
manojaarya TSR · Sep 8, 2025, 1:46 AM · Private Reports



Add Widget

Report: Purchase Orders based on Supplie

□ Use chart settings from report

Display As

Purchase Orders based on Suppliers

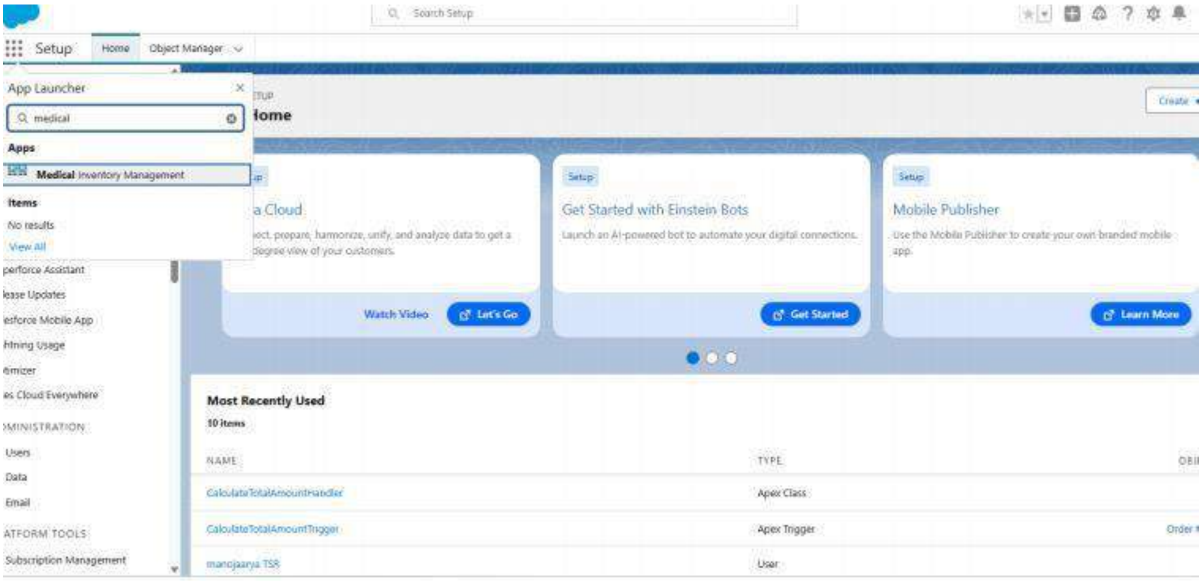
Activity 2: View Dashboard

Click on App Launcher (left-hand side of the screen).

Search for Medical Inventory Management → Click to open the app.

Go to the Dashboard tab.

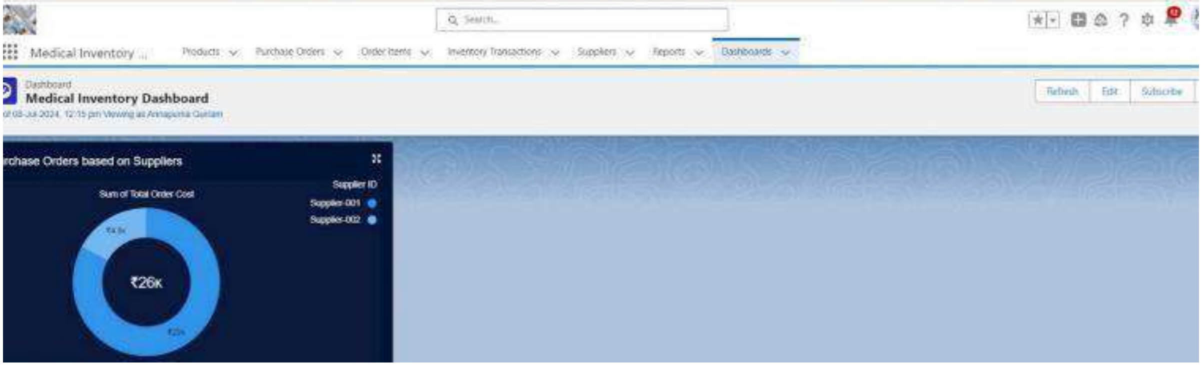
Click on Medical Inventory Dashboard to view the graphical representation of records.



Most Recently Used

10 items

NAME	TYPE	OBJ
CalculateTotalAmountHandler	Apex Class	
CalculateTotalAmountTrigger	Apex Trigger	Order I
manojaarya TSR	User	



Conclusion

The Medical Inventory Management System effectively automates and streamlines inventory management in a healthcare environment. By leveraging Salesforce CRM features, the system improves efficiency, ensures data accuracy, and enhances transparency in managing medical supplies. This project highlights the practical application of Salesforce in addressing real-world challenges, as part of the Naan Mudhalvan initiative.